

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Reconnaissance automatique de liens dans une page WEB : identification et représentation

Baldi, Julien

Award date:
1997

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



Facultés Universitaires Notre-Dame de la Paix

INSTITUT D'INFORMATIQUE

rue Grandgagnage, 21 5000 NAMUR

Tel 081/72.49.83. fax 081/72.49.67.

Année académique 1996-1997

**Reconnaissance automatique
de liens dans une page WEB :**
identification et représentation

Par

Julien BALDI

Mémoire présenté en vue de l'obtention du grade de
Maître en Informatique

<p>Reconnaissance automatique de liens dans une page WEB : identification et représentation</p>

Auteur : Julien BALDI

Promoteur : Madame Monique NOIRHOMME
Institut d'Informatique, FUNDP, Namur, Belgique

Maître de stage : Monsieur Louis-Claude PAQUIN
Université du Québec à Montréal, Montréal, Canada

Lieu de stage : Université du Québec à Montréal, Département des
Communications, laboratoire des technologies interactives
Pavillon Judith-Jasmin, 405, rue Sainte-Catherine Est,
H2L 2C4, Montréal, Canada

Résumé

Dans ce mémoire, nous traitons tout d'abord des langages de programmation et de création des pages WEB que sont JAVA et HTML. Ensuite, nous traitons de l'application ChercheWeb et des modifications que l'on peut apporter. Cette application de parcours des pages WEB et de reconnaissance des liens a été créée en Java lors d'un stage à l'Université du Québec à Montréal. Enfin, nous traitons des liens que l'on peut trouver sur le World Wide Web : nous en donnons une typologie, des méthodes de représentation et nous proposons quelques améliorations sur la manière de fournir un complément d'information sur les liens.

Abstract

In this paper, we discuss JAVA and HTML, two languages for programming and designing WEB pages. Then, we discuss the ChercheWeb application and the modifications we can bring. This application for browsing WEB pages and recognising the links was developed in Java at the University of Quebec in Montreal. Finally, we discuss the links available on the World Wide Web : we give a typology, some representation methods and we suggest some improvements on the way to provide more information about the links.

Avant Propos

L'élaboration d'un mémoire est une tâche difficile et il ne serait pas possible de la réaliser sans aide. Dans cet avant propos, nous tenons à remercier toutes les personnes qui nous ont assisté et conseillé, tout d'abord au cours du stage au Canada et ensuite lors de la rédaction de ce mémoire.

Au Canada, nous souhaitons remercier

Monsieur Jean-Pierre Boyer, Directeur du Département des Communications, Monsieur Louis-Claude Paquin, Directeur de la Maîtrise en Communications et notre Maître de Stage, Monsieur Dany Beaupré, animateur pédagogique au Laboratoire de Technologies Interactives ainsi que les étudiants de Maîtrise en Communication.

A l'Institut d'Informatique, nous souhaitons remercier

Madame Monique Noirhomme, notre promoteur et Messieurs Vincent Serpe et Manuel Rouard, chercheurs dans l'équipe de Madame Noirhomme.

Nous remercions également toutes les personnes qui ont effectué les différentes relectures et corrections de ce mémoire.

Table des matières

AVANT PROPOS	1
TABLE DES MATIERES	3
TABLE DES ILLUSTRATIONS	7
INTRODUCTION	9
CHAPITRE I : APERÇU DU LANGAGE JAVA	11
0. INTRODUCTION	11
1. HISTORIQUE	12
2. CARACTERISTIQUES	13
2.1. SIMPLE	13
2.2. ORIENTE-OBJET	14
2.3. DISTRIBUE	15
2.4. ROBUSTE	15
2.5. SUR	15
2.6. ARCHITECTURALEMENT NEUTRE	16
2.7. PORTABLE	16
2.8. INTERPRETE	16
2.9. PERFORMANT	17
2.10. MULTITHREAD	17
2.11. DYNAMIQUE	17
3. LIBRAIRIES DE BASE	18
4. DIFFERENCES ENTRE LES APPLETS ET LES APPLICATIONS	20
5. SECURITE EN JAVA	23
5.1. MECANISMES DE SECURITE	24
5.2. NOUVEAUX MECANISMES DE SECURITE	28
6. OUTILS ET AVENIR DE JAVA	29
7. AVANTAGES ET INCONVENIENTS	32
7.1. AVANTAGES	32
7.2. INCONVENIENTS	33
8. CONCLUSION	34

CHAPITRE II : HTML, LANGAGE DE CREATION DES PAGES WEB	35
0. INTRODUCTION	35
1. DETAIL DU MARQUAGE	36
1.1. MARQUAGE MINIMAL	36
1.2. MARQUAGE STRUCTUREL	37
1.3. STYLES	39
1.4. LISTES	39
1.5. TABLES	41
1.6. LIENS	42
2. CONCLUSION	47
CHAPITRE III : L'APPLICATION CHERCHEWEB	49
0. INTRODUCTION	49
1. ENONCE	49
2. PROBLEMES POSES ET SOLUTIONS APPORTEES	50
3. L'APPLICATION TERMINEE : CHERCHEWEB	51
3.1. LE MENU "FICHIER"	51
3.2. LE MENU "HISTORIQUE"	53
3.3. LE MENU "?"	53
4. AMELIORATIONS DE L'APPLICATION	54
5. UTILISATIONS POSSIBLES	55
6. CONCLUSION	56

CHAPITRE IV : LES LIENS	57
0. INTRODUCTION	57
1. TYPOLOGIE DES LIENS HYPERMEDIAS	57
1.1. NATURE	58
1.2. POSSIBILITE D'IDENTIFICATION AUTOMATIQUE	58
1.3. CARACTERISTIQUES DE DESTINATION	58
1.4. CARACTERISTIQUES PHYSIQUES	59
2. LIENS EXISTANT SUR LE WEB	59
3. REPRESENTATION DE LA STRUCTURE DES LIENS	61
3.1. LES CARTES EN DEUX DIMENSIONS	61
3.2. LES METAPHORES	64
4. EXAMEN D'UNE ADRESSE WEB	66
4.1. PREFIXE DE L'ADRESSE	66
4.2. SERVEUR	67
4.3. SUFFIXE DE LA PAGE	67
5. EXAMEN DE PAGES WEB	68
5.1. SECTEUR DES BANQUES	68
5.2. SECTEUR DE L'IMMOBILIER	71
5.3. SECTEUR DES FOURNISSEURS D'ACCES A INTERNET	72
5.4. OBSERVATIONS COMMUNES AUX TROIS SECTEURS	73
6. PRESENTATION ACTUELLE DES LIENS	74
7. PROPOSITIONS POUR UNE PRESENTATION	75
8. CONCLUSION	79
CONCLUSION	81
GLOSSAIRE	83
REFERENCES	93
ANNEXES	99

Table des Illustrations

Tableau I.1: Evolution des librairies, classes et interfaces du kit de développement	19
Figure I.1 : La nouveauté de JAVA	20
Figure I.2 : Les différentes façons d'exécuter un programme JAVA.	21
Tableau I.2 : Evolution des possibilités des programmes JAVA en fonction du kit de développement.....	23
Figure I.3 : Les mécanismes de sécurité de JAVA	26
Figure I.4 : Processus de création et d'exécution d'une applet.	28
Figure II.1 : Effet des marquages de styles	39
Figure III.1 : La boîte de dialogue "Rechercher..."	52
Figure V.1 : Une des premières représentations par Navigational View Builder	62
Figure V.2 : Icônes représentant les types de document	62
Figure V.3 : Visualisation au moyen de l'Hyperbolic Browser.....	63
Figure V.4 : Représentation au moyen d'objets 3D.....	64
Figure V.5 : Représentation WEBBook.....	65
Figure V.6 : Représentation du BookForager	65
Figure V.7 : Page WEB comportant des frames	69
Figure V.8 : Fenêtre de navigation de la Bacob.....	70
Figure V.9 : Exemple de demande de plug-in.....	73
Figure V.10 : Exemples d'icônes.....	76

Introduction

Au cours d'un stage réalisé au Département des Communications de l'Université du Québec à Montréal, il nous a été demandé de réaliser une application qui permettrait de parcourir des pages WEB et d'en retirer automatiquement les liens. L'application, réalisée en JAVA, devait également permettre de se rendre compte des caractéristiques du langage ainsi que de ses possibilités. Pour nous aider dans le développement de l'application, nous disposions de quelques ouvrages de référence et d'informations contenues sur Internet, notamment sur les sites de Sun et de JavaSoft qui sont, respectivement, les sites de la société ayant développé le langage et de la société commercialisant les outils de développement de JAVA.

Une fois l'application terminée, au retour du stage, nous nous sommes intéressés à la reconnaissance des liens existants dans les pages WEB. Nous avons essayé de comprendre en quoi consiste un lien, comment le reconnaître et représenter une structure de liens, quelles informations retirer d'un lien et comment étendre les informations que l'on peut retirer d'un lien.

Nous présentons le résultat de ces recherches en trois parties. Dans la première partie, nous traitons des langages de programmation et de création de pages du World Wide Web que sont JAVA et HTML. Dans la deuxième partie, nous présentons l'application réalisée qui porte le nom de ChercheWeb et nous proposons quelques améliorations. Dans la troisième et dernière partie, nous traitons des liens, nous en donnons une typologie, quelques moyens de représentation, nous voyons comment tirer de l'information de tels liens. Enfin, nous terminons par une conclusion récapitulant les axes de travail.

Première partie :
Les Langages de Programmation et
de création de pages du WWW

Chapitre I :
Aperçu du Langage JAVA

0. INTRODUCTION

Dans le présent chapitre, nous traitons du langage JAVA, utilisé pour le développement de l'application ChercheWeb (cfr. Chapitre III). Nous essayons d'en donner un aperçu le plus complet possible sans (trop) nous attarder aux aspects techniques de programmation. Ainsi, nous développons successivement l'historique (section 1), les caractéristiques (section 2) et les bibliothèques de base du langage (section 3). Nous expliquons les différences existant entre les deux types de programmes JAVA à savoir les applets et les applications (section 4). Nous évoquons les aspects de sécurité (section 5) et nous disons un mot sur les outils et sur l'avenir possible de JAVA, notamment sur les derniers développements du langage. Nous essayons de découvrir les avantages et les inconvénients du langage (section 7) et nous en tirons une conclusion (section 8).

Nous tenons à donner une explication sur la terminologie que nous emploierons concernant les programmes JAVA. Etant donné l'amalgame possible entre le terme application au sens informatique général et le terme application dans le contexte de JAVA, nous utiliserons le terme "programme JAVA" lorsqu'il s'agira aussi bien d'une applet que d'une application JAVA et le terme application ne sera utilisé que lorsqu'il s'agira d'une application dans le contexte JAVA.

1. HISTORIQUE

En 1990, la société d'informatique Sun était en crise. Patrick Naughton, un ingénieur, voulait quitter l'entreprise pour aller réaliser des travaux plus intéressants chez Next. Avant son départ, Scott McNealy, président de Sun, lui demanda de rédiger un rapport sur les raisons de la crise. Naughton rédigea ce rapport et McNealy le transmit à toute la partie management de l'entreprise. Des centaines de réponses lui parvinrent, toutes en accord avec le rapport de Naughton.

Afin de surmonter les problèmes, une équipe, du nom de code Green, fut constituée pour définir les principes d'un nouveau projet :

- conserver des équipes de projets assez petites pour se réunir facilement ;
- aller à la rencontre des besoins du consommateur ;
- simplifier l'environnement informatique ;
- créer du hardware simple et intuitif, utilisable par n'importe qui.

McNealy donna à cette équipe tous les pouvoirs ; les membres du projet étaient libres de faire ce qu'ils voulaient. Au cours de leurs recherches, ils se rendirent compte que la technologie informatique était présente dans de nombreux objets utilisés quotidiennement, tels que les fours, les magnétoscopes, les ascenseurs, mais qu'il n'existait encore aucune interface commune. L'idée fut alors de créer une interface commune pour faire tout fonctionner à l'aide d'un seul appareil et également améliorer les performances.

Pour piloter cet appareil, il fallait un langage de programmation. Le C++ n'étant pas assez fiable et ne permettant pas une interface accessible à tous, ils décidèrent de créer un nouveau langage. C'est ainsi que le langage Oak vit le jour en août 1991. Un an plus tard, l'équipe présenta sa réalisation : un appareil tenant dans la main, sans clavier ni bouton, mais disposant d'un écran tactile, permettant de piloter les objets électroniques domestiques. Toutefois, la commercialisation de ce produit fut un échec, car la production était trop onéreuse.

Sun, à nouveau en crise, décida alors de s'orienter vers un marché en pleine expansion : celui de la création de produits d'interface pour Internet. Ils rebaptisèrent Oak en JAVA et décidèrent d'en faire un standard en le distribuant gratuitement. Le lancement officiel eut lieu le 23 mai 1995.

2. CARACTERISTIQUES

Le langage de programmation JAVA fut créé pour résoudre de nombreux problèmes qui se posent, dans la programmation moderne, pour développer un environnement fiable, portable, distribué et travaillant en temps réel.

Ainsi, JAVA est un langage simple, orienté-objet, distribué, robuste, sûr, architecturalement neutre, portable, interprété, performant, multithread et dynamique¹.

2.1. SIMPLE

JAVA est simple : il est conçu pour être utilisé sans nécessiter une longue période de formation. Pour cela, les concepteurs ont créé un langage aussi proche que possible du C++, le langage le plus utilisé actuellement.

Pour le rendre plus familier, plus simple et plus facile à apprendre, ils ont supprimé quelques caractéristiques du C++ qui apportaient plus de problèmes que de bénéfices. Ainsi, ils ont enlevé le pré-processeur, l'héritage multiple, l'arithmétique des pointeurs et les tableaux multidimensionnels. Afin de faciliter la gestion de la mémoire, ils ont ajouté le mécanisme de "Auto Garbage Collection" qui, en libérant périodiquement la mémoire des objets qui ne sont plus utilisés, non seulement, rend la tâche de programmation plus facile, mais également réduit le nombre d'erreurs.

Un autre aspect de la simplicité de JAVA est sa taille : l'environnement d'exécution (le JRE), ne nécessite que 2.5 MB et ne nécessite pas de machine très puissante pour fonctionner.

¹ Basé sur [Sun95] et [Gosling]

Les programmes JAVA sont de deux types :

- les applets nécessitant un contexte (soit l'appletviewer, soit un navigateur, en anglais browser) pour s'exécuter ;
- les applications pouvant s'exécuter à l'aide d'un interpréteur indépendamment d'un navigateur.

Nous traiterons des principales différences entre les applets et les applications dans la section 4 "Différences entre les applets et les applications".

2.2. ORIENTE-OBJET

JAVA est un langage orienté-objet, non pas pour suivre la tendance générale en programmation, mais parce que l'architecture orienté-objet satisfait bien aux besoins de la technologie client-serveur et aux logiciels distribués.

Pour être considéré comme orienté-objet, JAVA supporte au moins ces quatre caractéristiques :

- encapsulation : utilisation de modules et "information hiding" ;
- polymorphisme : une méthode d'un même nom peut être déclarée avec des arguments de types différents ;
- héritage : des classes sont basées sur d'autres classes et peuvent utiliser leurs méthodes ;
- liens dynamiques : les objets peuvent venir de n'importe où et on peut utiliser ces objets sans connaître leur code spécifique.

Les objets sont des modèles pour la programmation. Dans la vie réelle, nous sommes entourés d'objets : les voitures, les télévisions, les chiens, etc. Les objets, prenons l'exemple de la voiture, ont des caractéristiques (couleur, vitesse maximale...) et des comportements (démarrer, tourner...). En programmation, les caractéristiques sont les variables d'instance et les comportements sont les méthodes. Celles-ci sont définies à l'intérieur de classes. Une classe n'est pas un objet mais un patron qui le définit. On peut donc créer un objet en instanciant une classe.

Contrairement à la plupart des langages orienté-objet, JAVA ne permet pas l'héritage multiple, c'est-à-dire qu'en JAVA, une classe ne peut hériter que d'une et une seule classe. Toutefois, il est possible d'utiliser les classes abstraites qui permettent de définir des méthodes pour les interfaces. Une interface étant une collection de définitions de méthodes (mais sans implémentation) et de déclarations de constantes.

2.3. DISTRIBUE

Les objets utilisés en JAVA peuvent se trouver sur des systèmes à distance ou sur des systèmes locaux. De plus, JAVA possède des bibliothèques permettant d'utiliser les protocoles TCP/IP comme HTTP et FTP.

2.4. ROBUSTE

La vérification des problèmes possibles se fait lors de la compilation plutôt que lors de l'exécution. Cette technique permet de repérer et de corriger les erreurs avant l'exécution et est donc plus robuste. De plus, le modèle de gestion de la mémoire de JAVA ne permet pas aux programmes d'accéder à l'entièreté de la mémoire. Ceux-ci ne peuvent donc pas modifier des données qu'ils ne sont pas autorisés à manipuler comme c'est le cas avec le C++, par exemple.

2.5. SUR

JAVA étant prévu pour être utilisé dans des environnements distribués, beaucoup de précautions ont été prises concernant la sécurité.

Des techniques d'authentification sont basées sur le chiffrement de clés publiques. De plus, on peut définir des accès aux classes en leur ajoutant les attributs suivants : public (toutes les classes ont accès à la classe publique), protected (seules les sous-classes ont accès à la classe protected), package (seules les classes du même package ont accès), et private (aucune autre classe n'a d'accès).

2.6. ARCHITECTURALEMENT NEUTRE

Avec JAVA, la même version d'un programme fonctionne sur toutes les plateformes sans changer le code. L'interpréteur JAVA permet cela en générant des instructions en byte-code, indépendantes de l'architecture particulière d'une machine. Ainsi, un programme JAVA n'est pas écrit pour une plate-forme particulière, il est prévu pour fonctionner sur une "Machine Virtuelle JAVA" : quand on exécute un programme JAVA, on lance un programme qui va simuler, sur la machine, le fonctionnement de la machine JAVA rendant ainsi l'exécution du programme possible. JAVA est donc indépendant des architectures matérielles et des systèmes d'exploitation.

2.7. PORTABLE

Le code JAVA est portable sur n'importe quel système où l'interpréteur et l'environnement d'exécution ont été implémentés. Pour ce faire, la taille des types de données de base a été spécifiée et standardisée.

2.8. INTERPRETE

L'interpréteur JAVA peut exécuter du code directement sur les machines sur lesquelles il a été implémenté. En fait, JAVA est à la fois compilé et interprété : le programmeur compile le code source pour obtenir un byte-code. Ce byte-code est alors interprété.

Comme l'interprétation se fait sur la plate-forme sur laquelle sera exécuté le programme, il n'est plus nécessaire de concevoir différentes versions du code source en fonction de la plate-forme.

2.9. PERFORMANT

Lorsque la performance apportée par le byte-code n'est pas suffisante (le code interprété est plus lent que le code compilé pour un système d'exploitation particulier), il est possible d'ajouter des programmes compilés dans d'autres langages que JAVA et les intégrer au programme JAVA. Ceci permet d'améliorer les performances, mais s'oppose à la portabilité car ces programmes compilés ne le sont que pour une plate-forme donnée.

2.10. MULTITHREAD

Avec le mécanisme des threads, il n'est pas nécessaire d'attendre qu'une tâche soit terminée pour en commencer une autre : plusieurs tâches peuvent s'effectuer "en parallèle". L'avantage de cette technique est un bon comportement à la fois interactif et temps réel. Remarquons toutefois que la capacité d'exécuter efficacement des tâches en parallèle dépend de la plate-forme sur laquelle le programme est porté. Les performances seront donc meilleures sur une plate-forme multiprocesseur que sur une plate-forme à un seul processeur.

2.11. DYNAMIQUE

Java a été conçu pour s'adapter à un environnement en constante évolution. Ainsi, dans les langages de programmation actuels, si une personne B utilise une librairie de classes créée par une personne A et que A change cette librairie, B devra recompiler ses programmes pour s'adapter à ce changement. Par contre, JAVA réalise une interconnexion entre les modules et leurs mises à jour, ce qui permet d'éviter ce problème. En outre, des méthodes et des variables peuvent être ajoutées ou supprimées sans nuire aux librairies de classes.

3. LIBRAIRIES DE BASE¹

Le système complet de JAVA comprend des bibliothèques de classes utilitaires, d'interfaces et de méthodes pour créer les programmes s'exécutant sur les différentes plateformes. Ces bibliothèques de base, pour le kit de développement JDK 1.0.2, sont :

- `java.applet` : qui fournit les classes et interfaces permettant de créer et utiliser les applets JAVA. De plus, l'interface `AppletContext` permet de connaître les informations concernant le contexte d'exécution de l'applet (que ce soit un navigateur WEB ou l'`appletViewer` de Javasoft).
- `java.awt` : qui fournit les classes et interfaces permettant de créer des interfaces fenêtrées (AWT est mis pour Abstract Windowing Toolkit).
Elle contient les outils interactifs concrets (boutons de commandes, listes de sélection, ...) et également la classe `Graphics` qui permet les dessins (carrés, cercles, ...) ainsi que l'affichage des images.
- `java.awt.image` : qui fournit les classes et interfaces permettant de traiter et de manipuler les images se trouvant soit sur le réseau soit sur la machine locale.
- `java.awt.peer` : qui fournit les classes et interfaces qui servent d'intermédiaire entre le code source et la machine sur laquelle le programme s'exécutera.
- `java.io` : qui fournit les classes et interfaces permettant de gérer les entrées et sorties en JAVA.
- `java.lang` : qui fournit les classes et interfaces de base du langage JAVA, elle définit les types (`Integer`, `String`, ...) et fournit les services de gestion de sécurité et de gestion du système.

¹ Basé sur [Campione] et [Newman]

- `java.net` : qui fournit les classes et interfaces permettant de communiquer à travers le réseau.
- `java.util` : qui constitue un ensemble de classes utilitaires ne pouvant pas être intégrées dans les autres librairies.

Dans la dernière version du kit de développement (JDK 1.1.3), la librairie `java.awt.peer` a été supprimée et les librairies suivantes ont été ajoutées :

<code>java.awt.datatransfer</code>	<code>java.awt.event</code>	<code>java.beans</code>
<code>java.lang.reflect</code>	<code>java.math</code>	<code>java.rmi.dgc</code>
<code>java.rmi</code>	<code>java.rmi.registry</code>	<code>java.rmi.server</code>
<code>java.security.acl</code>	<code>java.security</code>	<code>java.security.interfaces</code>
<code>java.sql</code>	<code>java.text</code>	<code>java.util.zip</code>

Ainsi l'évolution du nombre de librairies et de classes entre la version 1.0.2 et la version 1.1.3 est présentée dans le tableau I.1 :

	JDK 1.0.2	JDK 1.1.3
Public <code>java.*</code> packages	8	22
Public <code>java.*</code> classes and interfaces	211	477

Tableau I.1: Evolution des librairies, classes et interfaces du kit de développement

4. DIFFERENCES ENTRE LES APPLETS ET LES APPLICATIONS

Comme tous les autres langages de programmation, JAVA fournit aux développeurs de quoi gérer les entrées sorties, les fenêtres, les graphiques et tout ce dont ils ont besoin pour développer et exécuter les programmes sophistiqués. Mais JAVA offre de plus une nouvelle capacité : celle d'écrire un programme une fois et de l'exécuter sur différentes plates-formes matérielles et logicielles sans devoir modifier le code. (cfr. Figure I.1).

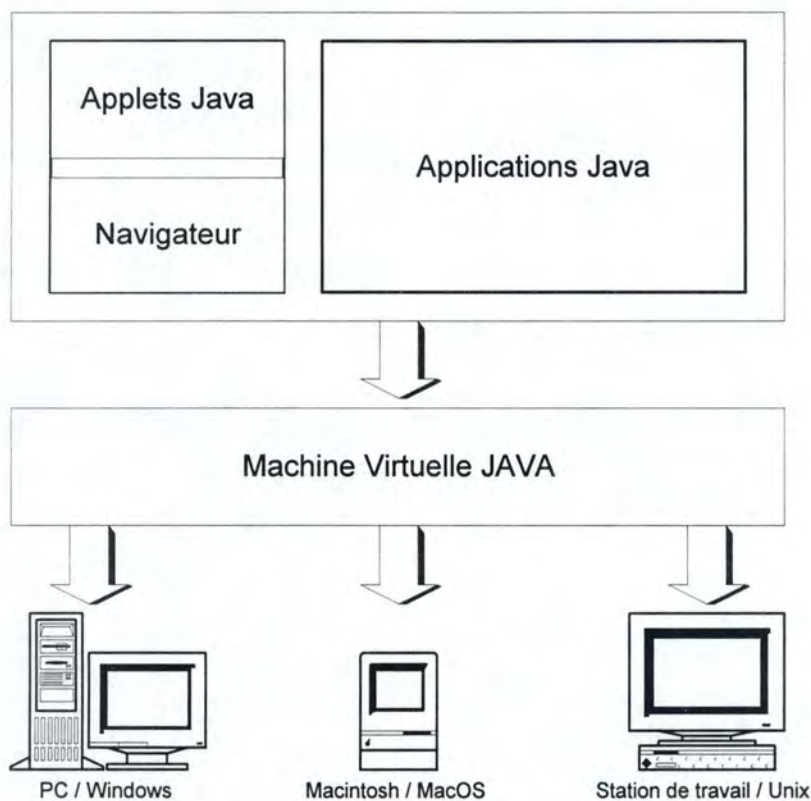


Figure I.1 : La nouveauté de JAVA

Cette capacité est due à la Machine Virtuelle JAVA. Celle-ci est un logiciel, implémenté au dessus du système d'exploitation de l'ordinateur sur lequel le programme sera exécuté, qui va simuler sur cet ordinateur le fonctionnement de la machine JAVA, rendant possible l'exécution du programme. La Machine Virtuelle JAVA peut donc fonctionner sur les ordinateurs et systèmes d'exploitation existant ou alors sur du matériel spécialement conçu pour JAVA, JavaOS.

Les programmes écrits en JAVA peuvent être de deux types : les applets, qui s'exécutent à l'aide d'un navigateur WEB, et les applications, qui s'exécutent indépendamment d'un navigateur WEB (cfr. Figure I.2).

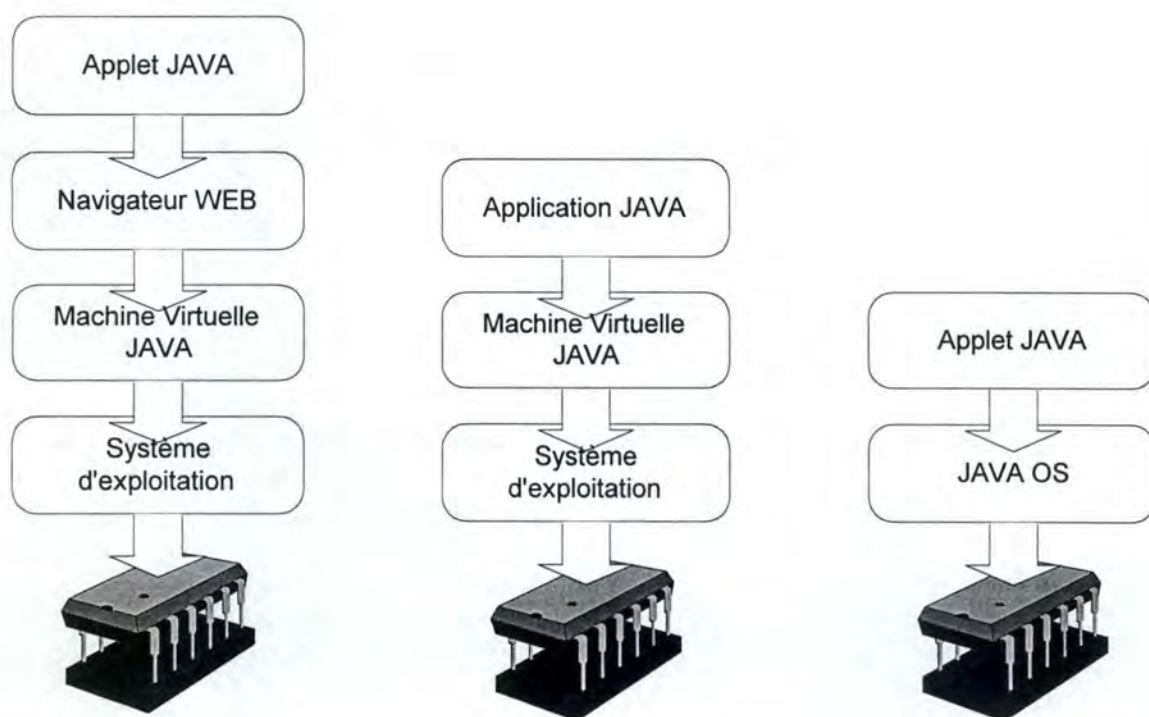


Figure I.2 : Les différentes façons d'exécuter un programme JAVA.

Les applets sont des programmes écrits en JAVA qui nécessitent un contexte pour s'exécuter. Les applications ne sont pas des programmes exécutables que l'utilisateur peut activer et utiliser directement, mais peuvent être exécutées directement grâce à l'interpréteur JAVA sans nécessiter de navigateur. L'environnement d'exécution charge une classe particulière et appelle la méthode "main()" de cette classe.

Comme nous l'expliquerons dans la section 5 "La sécurité en JAVA", les applets sont soumises à certaines restrictions pour des raisons de sécurité :

- elles ne peuvent pas charger des bibliothèques ou définir des méthodes natives ;
- normalement, elles ne peuvent lire ou écrire des fichiers sur la machine sur laquelle elles s'exécutent ;
- elles ne peuvent établir de connexions que sur le réseau d'où elles proviennent ;
- elles ne peuvent exécuter un programme sur la machine sur laquelle elles s'exécutent ;
- elles ne peuvent lire certaines propriétés du système.

Les applications, elles, ne sont pas soumises à ces restrictions. Toutefois, les applets possèdent des possibilités que les applications n'ont pas. Cela est dû aux interfaces et méthodes de la bibliothèque `java.applet`, notamment `AppletContext`, `AppletStub` et `AudioClip` :

- elles peuvent jouer des sons ;
- elles peuvent faire afficher des pages WEB au navigateur où elles s'exécutent ;
- elles peuvent utiliser les méthodes déclarées `public` des autres applets se trouvant sur la même page WEB ;
- les applets chargées sur le système de fichiers local n'ont pas les mêmes restrictions que les applets provenant du réseau.

L'évolution du Kit de développement JAVA a contribué à réduire ces différences, comme le montre le tableau I.2¹ :

Applets vs applications Version du JDK	Lire et écrire des fichiers locaux	Connexion réseau	Appel méthodes natives	Signature du code
Applets JDK 1.02	Non	Non	Non	Non
Applications JDK 1.02	Oui	Oui	Oui	Non
Applets JDK 1.1	Au choix du navigateur WEB	Au choix du navigateur WEB	Au choix du navigateur WEB	Oui
Applications JDK 1.1	Oui	Oui	Oui	Non
Applets futur JDK	Au choix de l'utilisateur	Au choix de l'utilisateur	Au choix de l'utilisateur	Oui
Applications futur JDK	Au choix de l'utilisateur	Au choix de l'utilisateur	Au choix de l'utilisateur	Oui

Tableau I.2 : Evolution des possibilités des programmes JAVA en fonction du kit de développement

5. SECURITE EN JAVA

La sécurité est une préoccupation de plus en plus importante à mesure que les produits et les services se répandent sur les réseaux en général et sur Internet en particulier. JAVA permet d'aller d'utiliser des programmes sur les réseaux, mais ces programmes sont-ils absolument sûrs ? Ne risquent-ils pas de causer des dommages aux machines sur lesquelles ils sont utilisés ? JAVA ne laisse-t-il pas la porte de nos disques durs ouverte aux programmes malveillants ?

¹ Tiré de [Byte 97b] et [Sun 1]

Les applets JAVA permettent d'obtenir de nouvelles fonctionnalités interactives dans les domaines des réseaux et de résoudre d'importants problèmes pour les architectures clients serveurs, mais elles soulèvent également de nouveaux problèmes de sécurité. En effet, alors qu'avec les techniques traditionnelles, il est possible de se protéger des actions malveillantes (par exemple grâce à des contrôles physiques et des firewalls), la capacité d'exécuter des applets en provenance du réseau augmente la vulnérabilité du système. Comme l'applet est importée par l'intermédiaire du navigateur de l'utilisateur et s'exécute localement, un logiciel peut voler ou endommager des informations stockées sur l'ordinateur.

5.1. MECANISMES DE SECURITE¹

Afin de se protéger des dangers exprimés ci-dessus, le langage JAVA met en œuvre plusieurs mécanismes :

- les mécanismes intégrés au langage ;
- les mécanismes de sécurité en tant que tels.

5.1.1. Les mécanismes intégrés au langage

Certains mécanismes sont liés aux caractéristiques du langage et offrent un certain niveau de sécurité :

- orienté-objet : les données sont masquées au sein des déclarations privées ;
- utilisation de classes finales : classes qui empêchent la modification de leur code ;
- absence de pointeurs et gestion de la mémoire automatique : les utilisateurs n'ont pas accès à la mémoire.

¹ Basé sur [Byte 97a], [Fritzinger]

5.1.2. Les mécanismes de sécurité en tant que tels

Mais il existe un système de gestion de la sécurité, plus performant, basé sur trois mécanismes :

- le vérificateur de byte-code ;
- le chargeur de classes ;
- le gestionnaire de sécurité.

Le processus de sécurité est décomposé en 6 étapes :

1. le navigateur WEB télécharge l'applet JAVA d'une page WEB ;
2. le "vérificateur de byte-code" effectue la vérification du byte-code ;
3. le "chargeur de classe" instancie l'applet dans un espace de noms ;
4. l'applet commence à s'exécuter et tente une méthode malveillante ;
5. le "gestionnaire de sécurité" est consulté avant l'exécution de la méthode ;
6. le "gestionnaire de sécurité" effectue ses vérifications et rejette la méthode malveillante.

La figure I.3 représente cette décomposition du processus de sécurité.

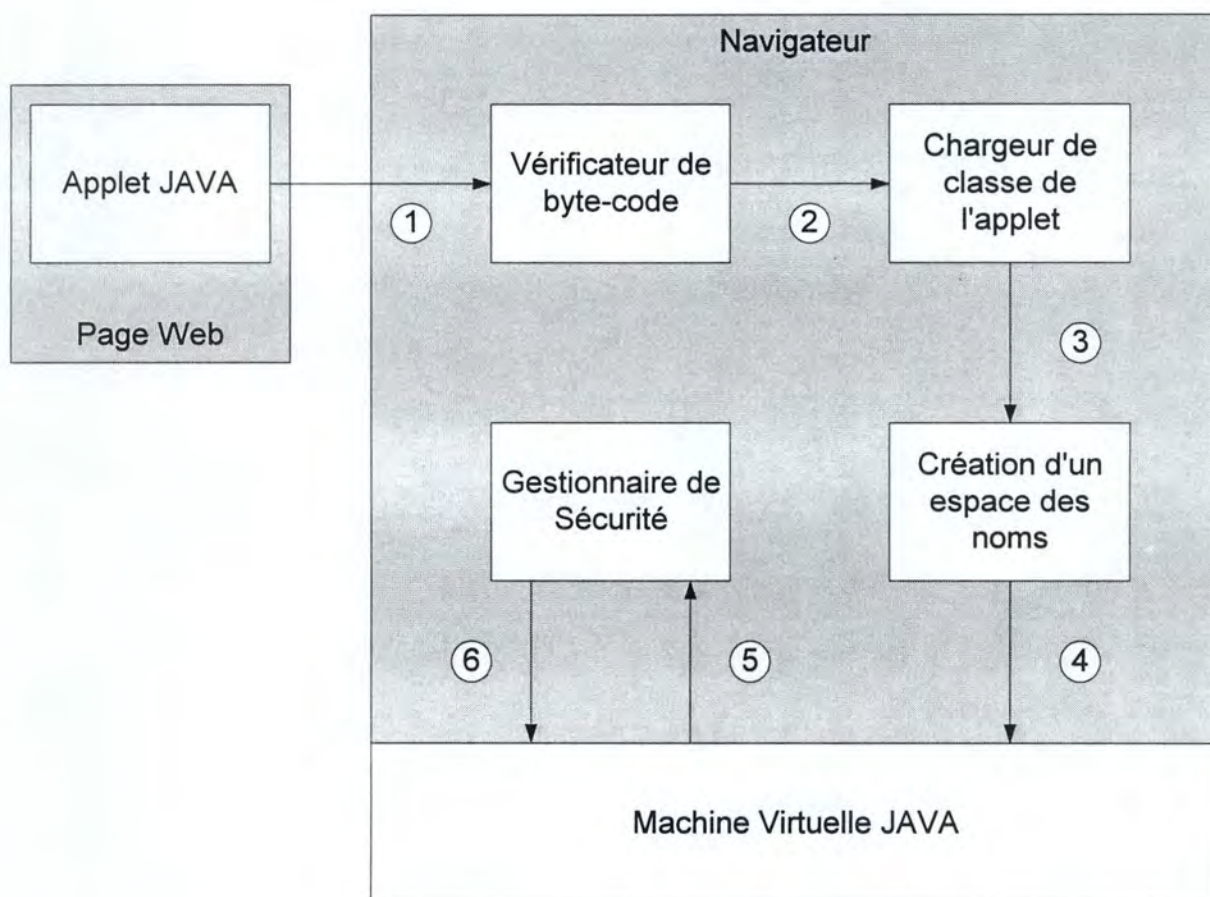


Figure I.3 : Les mécanismes de sécurité de JAVA

Voyons à présent ce mécanisme plus en détail.

A l'intérieur du navigateur, les actions des applets sont limitées à des zones où elles peuvent faire ce qu'elles veulent, appelées "sandboxes". En dehors de ces zones, il leur est interdit de lire ou de modifier des données. La sandbox comprend un certain nombre de composants des systèmes coopératifs, allant du gestionnaire de sécurité, qui s'exécute comme une partie du programme, aux mesures de sécurité comprises dans la Machine Virtuelle JAVA et dans le langage lui-même. Cette sandbox assure qu'un programme non sûr, voire malveillant, n'accède pas aux ressources du système.

Lorsqu'une applet est importée via le réseau, le navigateur fait appel au chargeur de classes. Celui-ci détermine quand et comment une applet peut charger des classes, charge le byte-code et crée, pour chaque applet, un espace de noms pour les classes importées. Cet espace de noms définit les portions de la Machine Virtuelle JAVA auxquelles peut accéder cette applet.

En maintenant un espace de noms séparé pour les applets locales, le chargeur de classes empêche que les applets importées n'aient accès à des zones privilégiées du système. De plus, elles ne peuvent ni créer leurs propres chargeurs de classes ni appeler des méthodes du chargeur de classes du système. Il faut toutefois noter deux remarques importantes. Primo, les navigateurs WEB ont leurs propres chargeurs de classes. Secundo, il est possible de créer une application avec son propre chargeur de classes. Dans les deux cas, si le chargeur de classes créé ne respecte pas les conventions de sécurité ci-dessus, il est possible de passer outre ce mécanisme.

Lorsqu'un navigateur importe du code, l'environnement d'exécution de JAVA le soumet au vérificateur de byte-code. Celui-ci vérifie que le code est conforme aux spécifications du langage et effectue une preuve de programme pour déterminer s'il ne crée pas de pointeur, s'il ne viole pas les restrictions d'accès et s'il ne change pas la nature des objets. Il s'assure donc que le byte-code pourra être passé à l'interpréteur sans risquer de le corrompre.

Lorsqu'une applet essaie d'effectuer une action qui risquerait de corrompre la machine locale ou d'accéder à des informations interdites, la Machine Virtuelle JAVA fait appel au gestionnaire de sécurité pour lui demander si l'action peut être effectuée correctement. Si c'est le cas, l'action est exécutée. Sinon, la Machine Virtuelle JAVA crée une exception de sécurité et affiche un message d'erreur.

On peut donc résumer les actions de la sandbox comme suit :

- seules les classes correctes sont chargées ;
- les classes non sûres ne peuvent exécuter des instructions dangereuses ;
- les classes non sûres n'ont pas accès aux ressources protégées du système.

Le processus complet de création et d'exécution de l'applet peut donc être représenté à la figure I.4.

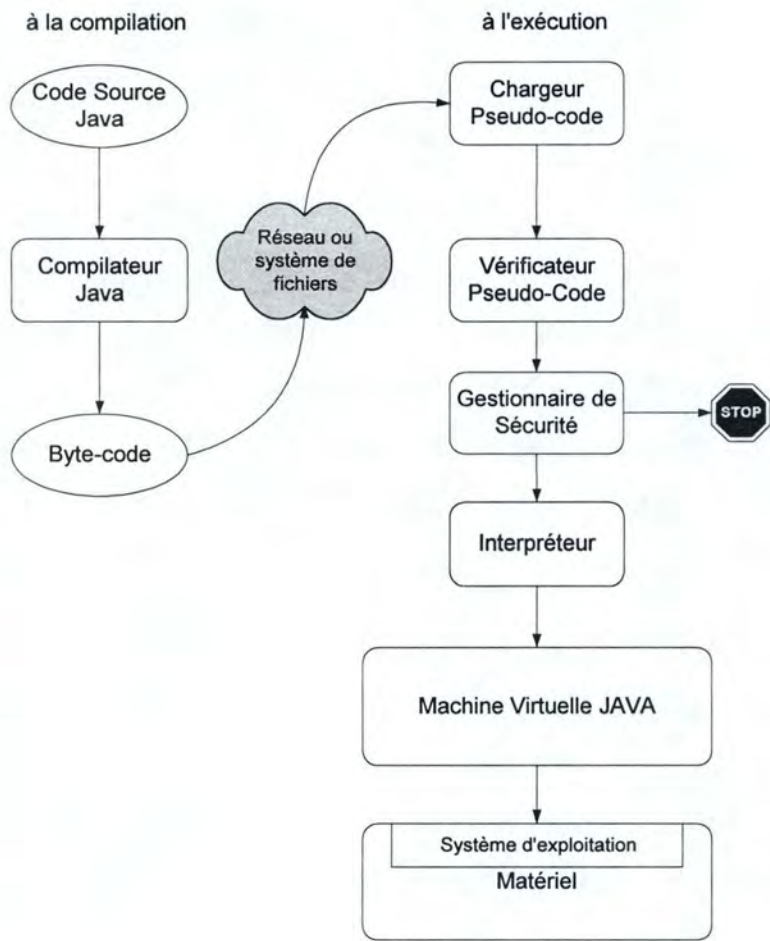


Figure I.4 : Processus de création et d'exécution d'une applet.

5.2. NOUVEAUX MECANISMES DE SECURITE

Le modèle de la sandbox ne protège l'utilisateur que contre les dommages causés par des actions malveillantes ou des virus. Toutefois, d'autres problèmes de sécurité se posent également. Ainsi, il est nécessaire d'utiliser des moyens d'authentification pour garantir que les applets proviennent bien d'où elles sont censées provenir. On pourrait alors décider que des applets venant de certains sites sont "sûres" et leur imposer moins de restrictions. De plus, les applets transitant sur le réseau ne sont pas chiffrées et toute personne malveillante pourrait les voler et ainsi avoir accès à des codes privés.

Les solutions proposées sont :

- les fichiers signés JAR pour JAVA Archive

Le programmeur enregistre le code de son applet et les fichiers qui s'y rapportent dans un fichier JAR. Ensuite, il crée une signature électronique basée sur ce fichier JAR. Les signatures non valides seront rejetées comme étant des fichiers corrompus.

- les logiciels de sentinelle

Ils enregistrent toutes les transactions qui se produisent sur le système. Lorsque qu'un problème se produit, il est possible d'en connaître la cause et ainsi de faire en sorte que cela ne se reproduise plus,

- le chiffrement

Il permet de cacher le code contenu dans les applets transitant sur le réseau.

6. OUTILS ET AVENIR DE JAVA

Petit à petit, des outils de programmation arrivent dans le commerce. Voici une liste non exhaustive des outils de développement pour JAVA :

- kit de développement de JAVA chez SUN ;
- SIMANTEC Café et Visual Café ;
- MICROSOFT Visual J++ ;

Les navigateurs WEB, quant à eux, sont dorénavant compatibles avec la technologie JAVA. Mais le navigateur JAVA par excellence reste HOTJAVA, créé par la société JAVASOFT en langage JAVA.

En ce qui concerne l'avenir de JAVA, il semble assuré puisque de grandes sociétés de logiciels, telles que Microsoft, IBM et APPLE, ont décidé, depuis le 30 avril 1996, d'intégrer JAVA dans leurs nouveaux produits. La société COREL fut la première à annoncer une suite bureautique écrite complètement en JAVA et MICROSOFT intégrera JAVA en standard dans sa nouvelle version de Windows.

Il y a quelques temps, la société américaine APPLIX a sorti Anywhere Office for JAVA, une suite bureautique pour environnement JAVA. Elle comprend un traitement de textes, un accès aux bases de données, un tableur, un logiciel de graphes et un logiciel de courrier électronique.

Les produits JAVA actuellement disponibles chez Javasoft¹ sont :

- JDK™ 1.1.3 : la dernière version du kit de développement de JAVA incluant les derniers développements de JAVA : JavaBeans, fichiers JAR, JDBC, JNDI, JFC ²...
- JRE 1.1.3 : la dernière version de l'environnement d'exécution permettant d'exécuter les programmes JAVA mais ne contenant pas les compilateurs, débogueurs et outils de développement.
- JavaBeans™ : une API indépendante de la plate-forme et de l'architecture pour créer et utiliser des composants dynamiques JAVA. Elle est supportée par les applets et les applications JAVA. Elle est destinée à s'intégrer dans les architectures de composants existantes telles que OLE et Active-X de Microsoft, LiveConnect de Netscape et OpenDoc.
- JFC : l'extension à l'AWT original de JAVA qui ajoute un ensemble de classes permettant d'améliorer l'interface graphique de JAVA. Le JFC contient un mécanisme de prise en charge des événements, d'opérations sans souris, de menus contextuels, d'impression, et de support de presse-papiers. Les prochaines versions du JFC comprendront des fonctions de Drag & Drop, des fonctions d'accessibilité pour les handicapés (support du braille, zoom de l'écran et lecture du texte), des fonctions pour les objets en deux dimensions et des objets interactifs similaires quelle que soit la plate-forme utilisée afin d'avoir une interface cohérente. Le JFC est livré avec les kits de développement JAVA à partir de la version JDK 1.1, et est compatible avec JavaBeans™.
- Security : l'API de sécurité permettant aux développeurs d'intégrer des fonctions de sécurité telles que la cryptographie, les signatures digitales, le chiffrement et l'authentification.
- JDK 1.0.2 : la version "originale" du kit de développement de JAVA.

¹ Basé sur [Sun 2], [Sun 3] et [Sun 4]

² Les abréviations sont expliquées dans la suite de ce mémoire et reprises dans le glossaire.

- HotJava™ : le navigateur écrit en JAVA.
- JavaOS : le système d'exploitation optimisé pour JAVA.
- JAVA Electronic Commerce : un terminal virtuel de paiement accessible par les navigateurs compatibles JAVA.
- JNDI™, fournissant un standard d'entreprise, une connectivité uniforme pour des informations industrielles permettant aux développeurs de créer des programmes JAVA avec des accès unifiés aux différents services de "naming and directory" au sein d'une entreprise.
- JAVA IDL, fournissant une interopérabilité avec CORBA (Common Object Request Broker Architecture), le standard industriel de développement de logiciels hétérogènes.
- JDBC™, fournissant une interface uniforme et des outils pour les bases de données relationnelles.
- Et aussi JAVA Server API, JAVA Media API, JAVA Management API.

7. AVANTAGES ET INCONVENIENTS

7.1. AVANTAGES

Les avantages principaux du langage JAVA sont les suivants :

- langage simple à apprendre et à utiliser (l'orienté objet est connu par beaucoup) ;
- portabilité : un seul code pour plusieurs plates-formes ;
- gestion automatique de la mémoire ;
- sécurité avancée (pour les versions les plus récentes de JAVA) ;
- utilisation de programmes à travers les réseaux (locaux, Intranet, Internet) ;
- réduction et gestion avancée des erreurs ;
- prix : JAVA est gratuit ;
- taille : nécessite peu d'espace disque (comparé aux autres environnements de développement) ;
- programmation "parallèle" grâce aux threads ;
- universalité : devient progressivement un standard de programmation dans les grandes firmes de logiciels.

Au vu de ces avantages, JAVA semble promis à un bel avenir, il est souvent le langage de programmation pour les nouvelles applications et notamment pour celles qui utilisent les réseaux. Il devient un concurrent sérieux des langages de programmation classiques (tels que le C et le C++) et pourrait même les remplacer d'ici peu.

7.2. INCONVENIENTS

Les principaux inconvénients du langage JAVA sont les suivants :

- restrictions et limites des applets et applications ;
- modèle de sécurité faible (pour les anciennes versions de JAVA) ;
- technologie encore immature (mais cela semble s'arranger avec les nouvelles versions) ;
- petits problèmes de compatibilité entre les différents kits de développement JAVA ;
- vitesse d'exécution : le code interprété est plus lent que le code compilé pour un système d'exploitation particulier ;
- interface AWT pauvre : manque d'Objets Interactifs, par exemple, les boîtes de combinaison (ce problème est résolu grâce au JFC) ;
- peu d'outils de développement, notamment pour la création d'interfaces (mais il semble que les sociétés d'informatique lancent leurs produits en ce sens).

Les quelques inconvénients présentés ci-dessus ne sont pas insurmontables. La preuve en est que dans les nouvelles versions du kit de développement JAVA la plupart de ces inconvénients n'existent plus. On peut donc s'attendre à voir JAVA se généraliser pour le développement de programmes. A moins bien sûr qu'il n'y ait d'autres inconvénients avec l'arrivée de nouvelles versions (la compatibilité entre les versions est un élément critique).

8. CONCLUSION

Depuis sa création en 1995, le langage JAVA a fortement évolué. Il est resté un langage de programmation facile à apprendre et à utiliser, offrant les avantages des langages orientés objets, et possédant de nouvelles fonctionnalités intéressantes telles que la capacité d'écrire un seul code pour plusieurs plates-formes et la gestion automatique de la mémoire. Les améliorations des mécanismes de sécurité, des possibilités d'interface graphique, des accès aux bases de données et des fonctions réseaux semblent avoir porté leurs fruits.

Les avantages de Java sont de plus en plus nombreux et les inconvénients ont tendance à disparaître avec l'arrivée de nouvelles versions des kits de développement. Cela entraîne que de plus en plus de sociétés de logiciels développent leurs produits en JAVA. Il devient ainsi capable de concurrencer (et même de supplanter?) les autres langages de programmation et notamment le C++.

Chapitre II :

HTML, Langage de Création des Pages WEB

0. INTRODUCTION¹

Dans le présent chapitre, nous traitons du langage HTML. Il est nécessaire de bien en connaître la syntaxe pour comprendre le fonctionnement de l'application ChercheWEB développée dans le chapitre III. En effet, l'application explore les pages WEB écrites en HTML pour en retirer les liens. Nous parcourons donc la syntaxe du langage en expliquant le marquage dans la section 1 et nous terminons par une conclusion dans la section 2.

HyperText Markup Language (HTML) est un langage de définition d'hypertexte standardisé par l'IETF (Internet Engineering Task Force). Les documents HTML permettent de présenter de l'hypertexte, des images, des sons.... HTML est utilisé depuis 1990 dans le cadre de l'initiative W3 du CERN et est une implémentation de SGML (Standard Generalized Markup Language).

En encadrant le texte voulu par des marques, on peut lui attribuer une présentation. Les marques sont délimitées par des symboles < et > et la balise de fin est la même que la balise de début mais précédée d'un symbole /. Le marquage du texte s'effectue donc comme suit :

texte normal <balise> texte affecté par la marque </balise> texte normal.

Il est important de noter que le marquage décrit ici s'applique à HTML 2 et HTML 3.x qui sont les standards actuels (HTML 4, qui vient de sortir, n'est pas encore intégré en standard dans les navigateurs). Certaines marques décrites ne sont toutefois pas reprises par les navigateurs alors que des marques ne faisant pas partie de HTML ont été intégrées (c'est le cas de la marque <APPLET> qui permet d'exécuter une applet JAVA dans une page WEB).

1. DETAIL DU MARQUAGE

Dans cette section, nous allons présenter le marquage du standard HTML. Nous ne traiterons pas du marquage le moins utilisé mais nous expliquerons en détail certaines marques car elles nous semblent importantes pour la compréhension du déroulement de l'application ChercheWEB qui sera développée dans la deuxième partie. Nous expliquerons la marque `<Applet>` qui ne fait pas partie du standard HTML mais qui a été ajoutée afin de permettre l'utilisation des applets JAVA dans les navigateurs WEB. Lorsque nous le jugerons utile, nous donnerons des exemples récapitulatifs comportant non seulement le code mais également le résultat obtenu.

Le but de ce chapitre n'étant pas de faire un cours exhaustif sur le langage HTML, mais de présenter le langage en vue d'une meilleure compréhension de l'application ChercheWEB, nous n'expliquerons pas l'utilisation des CGI, qui permettent de créer des formulaires, ni des frames, qui permettent de découper l'écran du navigateur en plusieurs documents HTML.

1.1. MARQUAGE MINIMAL

Les documents HTML doivent impérativement commencer par la marque `<HTML>` et se terminer par la marque `</HTML>`. Le titre du document HTML est identifié par le marquage `<HEAD>` et `</HEAD>`. On identifie le titre du document HTML au sein du navigateur au moyen du marquage `<TITLE>` et `</TITLE>`. La différence avec le marquage précédent est que `<HEAD>` indique le titre sur la page et que `<TITLE>` indique le titre dans la barre de titre du navigateur (c'est ce titre qui sera repris pour la gestion des signets ou bookmarks).

Tout le reste du texte du document HTML doit se trouver entre `<BODY>` et `</BODY>`. Les commentaires, qui ne seront pas affichés par le navigateur, sont placés entre le marquage `<!--` et `-->`.

¹ Basé sur [Berners-Lee], [CWRU], [Graham], [NCSA], [Sandia]

1.2. MARQUAGE STRUCTUREL

Le marquage suivant est utilisé pour structurer le document HTML :

1.2.1. <Hn> et </Hn>

Pour créer une hiérarchie de titres et de sous-titres, on place le marquage <Hn> et </Hn> où n vaut de 1 à 6, <H1> étant le niveau de titre le plus haut. A chaque niveau, il y a changement de la taille de la police de caractères et changement de paragraphe.

Exemple :

```
<H1> Voici un titre de niveau 1 </H1>
<H2> Voici un titre de niveau 2 </H2>
<H3> Voici un titre de niveau 3 </H3>
<H4> Voici un titre de niveau 4 </H4>
<H5> Voici un titre de niveau 5 </H5>
<H6> Voici un titre de niveau 6 </H6>
```

Voici le résultat affiché :

Voici un titre de niveau 1
Voici un titre de niveau 2
Voici un titre de niveau 3
Voici un titre de niveau 4
Voici un titre de niveau 5
Voici un titre de niveau 6

1.2.2.

La marque
 permet de passer à la ligne.

1.2.3.

La marque <P> permet de délimiter les paragraphes d'un texte en terminant la ligne et en insérant une nouvelle.

1.2.4. <HR>

La marque <HR> effectue un passage à la ligne et trace un ligne horizontale sur toute la largeur du document.

1.2.5. <TAB>

La marque <TAB> permet d'insérer des tabulations dans un document.

1.2.6. <ADDRESS> et </ADDRESS>

On peut inclure l'adresse et les coordonnées de l'auteur de la page WEB en les plaçant entre le marquage <ADDRESS> et </ADDRESS>. Les sauts de lignes dans l'adresse doivent se faire au moyen de la marque
.

1.2.7. <BANNER> et </BANNER>

Si une partie du document doit toujours être visible lorsque la page défile, on place cette partie entre les marques <BANNER> et </BANNER>. Ce marquage doit se trouver juste après <BODY>. Ce marquage intéressant n'est pas supporté par le navigateur Netscape.

1.2.8. <BLOCKQUOTE> et </BLOCKQUOTE>

Le marquage <BLOCKQUOTE> et </BLOCKQUOTE> permet de créer un bloc pour des citations, celles-ci seront mises en retrait.

1.2.9. <PRE> et </PRE>

On utilise la marque <PRE> pour définir une section du document où on tient compte des espaces et des passages à la ligne.

1.2.10.<FN> et </FN>:

Le marquage <FN> et </FN> est utilisée pour ajouter un pied de page à un document. Ce marquage intéressant n'est pas supporté par le navigateur Netscape.

1.3. STYLES

Le tableau II.1 reprend les différents styles applicables aux caractères :

Marque	Effet	Code	Exemple
	affiché en gras	avant texte après	avant texte après
<I>	affiché en italique	avant <I> texte </I> après	avant <i>texte</i> après
<U>	affiché en souligné	avant <U> texte </U> après	avant <u>texte</u> après
<STRIKE>	affiché en barré	avant <STRIKE> texte </STRIKE>après	avant texte après
<CODE>	police non proportionnelle	avant <CODE> texte </CODE> après	avant <code>texte</code> après
<BIG>	affiché plus grand	avant <BIG> texte </BIG> après	avant <big>texte</big> après
<SMALL>	affiché plus petit	avant <SMALL> texte </SMALL> après	avant <small>texte</small> après
<SUB>	affiché en indice	avant _{texte} après	avant _{texte} après
<SUP>	affiché en exposant	avant ^{texte} après	avant ^{texte} après

Figure II.1 : Effet des marquages de styles

1.4. LISTES

On peut définir une liste non numérotée en plaçant la marque de départ , puis on place avant chaque élément de la liste le marquage et on termine la liste avec la marque de fin . Le marquage et permet de définir des listes ordonnées et s'utilise de la même manière que les listes non numérotées. On utilise le marquage <DL> et </DL> de la même manière que les listes non numérotées pour établir une liste définition (<DD>) de mot ou de terme (<DT>).

Exemple :

```
<UL>
<LI> un élément;
<LI> un autre élément;
<LI> encore un autre élément;
</UL>

<OL>
<LI> un élément de liste ordonnée;
<LI> un autre élément de liste ordonnée;
<LI> encore un autre élément de liste ordonnée;
</OL>

<DL> 1er niveau de liste : exemple de DL
  <DT> exemple de DT
  <DT> encore un exemple de DT
    <DL> on change de niveau avec un DL
      <DL> encore une fois avec un DL
        <BR> on passe à la ligne avec un BR
      </DL> on revient au niveau précédent avec un /DL
    </DL>
  </DL>
Revoici le texte normal
```

Remarque : l'indentation n'est pas nécessaire, mais elle permet de mieux comprendre la structure.

Voici le résultat :

- un élément;
 - un autre élément;
 - encore un autre élément;
-
1. un élément de liste ordonnée;
 2. un autre élément de liste ordonnée;
 3. encore un autre élément de liste ordonnée;

```
1er niveau de liste : exemple de DL
exemple de DT
encore un exemple de DT
  on change de niveau avec un DL
    encore une fois avec un DL
      on passe à la ligne avec un BR
    on revient au niveau précédent avec un /DL
```

Revoici le texte normal

1.5. TABLES

On utilise le marquage <TABLE> et </TABLE> pour définir une table. Pour ajouter des cellules, on utilise le marquage <TD> et </TD>, pour ajouter des colonnes, on utilise le marquage <TR> et </TR>. On peut définir le contour des colonnes en ajoutant la marque <TABLE BORDER>. On peut ajouter des entêtes aux tableaux en utilisant la marque <TH>. Le texte inclus dans ces cellules est mis en gras. En utilisant l'attribut COLSPAN=nombre avec le marquage <TH> et <TD>, il est possible de créer des cellules qui sont plus larges qu'une colonne. En utilisant l'attribut ROWSPAN=nombre, il est possible de créer des cellules qui sont plus larges qu'un rang.

Exemple :

```
<TABLE BORDER>
<TH ROWSPAN=2> une colonne </TD>
<TH> une autre colonne </TD>
<TH> encore une autre colonne </TD>
<TR>
<TD> une autre colonne </TD>
<TD> encore une autre colonne </TD>
<TR>
<TD> et une colonne </TD>
<TD> et une autre colonne </TD>
<TD> et encore une autre colonne </TD>
</TABLE>
```

Le résultat est le suivant :

une colonne	une autre colonne	encore une autre colonne
	une autre colonne	encore une autre colonne
et une colonne	et une autre colonne	et encore une autre colonne

1.6. LIENS

Les liens permettent de se rendre d'un document WEB à un autre (qu'il s'agisse d'une page HTML, d'un fichier audio, d'une séquence vidéo, ou d'un fichier à télécharger) au moyen d'un simple clic de la souris. Il se peut que le navigateur ne puisse pas utiliser directement le lien. En effet, lorsqu'il s'agit d'un lien vers un type de fichier qu'il ne connaît pas, il propose à l'utilisateur trois possibilités : soit de sauvegarder le fichier inconnu, soit de lancer une application ou un plug-in, soit de ne rien faire. Un plug-in est un composant facultatif du navigateur permettant d'utiliser un certain type de fichier (par exemple un fichier realaudio de radio en temps réel).

Au niveau de la présentation, le lien ressort généralement du reste du texte en étant souligné et affiché dans une autre couleur. De plus, lorsque le curseur de la souris passe sur un lien, il change de forme (le plus souvent, le curseur devient une petite main). Enfin, signalons qu'on peut attacher un lien à du texte ou à une image.

1.6.1. et

Le marquage `<A>` et `` permet de définir un lien vers un autre document HTML (sur le même serveur ou sur un autre serveur), un fichier multimédia ou une section de référence dans le document courant et s'utilise seulement dans le corps du document.

Lien vers un document sur un autre serveur

On peut créer des liens vers d'autres serveurs comme ceci :

```
<A HREF="http://www.autre_serveur.suffixe/document.html"> Document </A>
```

Lien vers un document sur le même serveur

Quand on veut créer des liens sur le même serveur que celui où se trouve le document actuel, on peut utiliser soit les liens absolus (qui contiennent le chemin d'accès complet), soit les liens relatifs (qui tiennent compte du répertoire dans lequel se trouve le document courant). En général, les liens relatifs sont préférables car la mise en page est plus rapide et la gestion de l'arborescence est facilitée.

Dans l'exemple ci-dessous, le premier lien est un lien absolu et le second un lien relatif.

```
<A HREF="http://www.mon_serveur.suffixe/sous_répertoire/document.html">
```

```
Document</A>
```

```
<A HREF="sous_répertoire/document.html">Document</A>
```

Lien vers une section de référence

Il est possible de lier du texte vers une autre section du même document. Pour ce faire on détermine le point d'ancrage avec l'étiquette **HREF** qui prend pour valeur le nom de la section cible; le signe "#" indiquant qu'il s'agit d'un lien vers une section précise du document. Ensuite, il faut identifier la section cible en utilisant l'attribut **NAME**.

Exemple :

```
Point d'ancrage : Pour plus d'informations, voir la <A  
HREF="#documentation"> Documentation </A>
```

```
Section cible : <A NAME="documentation"> Documentation </A>
```

Il est également possible de créer une référence à une section dans un autre document, il suffit dès lors d'ajouter avant le "#" le lien de cet autre document.

```
Exemple :  
Pour plus d'informations, voir la <A HREF="document.html#documentation">  
Documentation </A>
```


1.6.2. et

Le marquage `` et `` permet d'inclure des images dans un texte. La syntaxe est similaire à celle du marquage `<A>` et ``.

L'attribut `SRC` permet d'indiquer la source de l'image. L'attribut `ALT` permet de substituer du texte à une image qui ne peut pas être affichée. L'attribut `ALIGN` permet d'aligner le texte par rapport à l'image, soit au dessus (`TOP`), soit au milieu (`MIDDLE`) soit en dessous (`BOTTOM`). L'attribut `ISMAP` permet d'indiquer que l'image est une carte c'est-à-dire que selon l'endroit où l'on clique, une page différente est référencée.

1.6.3. <FIG> et </FIG>

Le marquage `<FIG>` et `</FIG>` permet un meilleur contrôle des images incorporées grâce à plusieurs attributs. L'attribut `NOFLOW` indique de ne pas afficher de texte aux côtés de la figure. Les attributs `WIDTH` et `HEIGHT` définissent la grosseur de l'image en pixel. L'attribut `UNITS` permet de définir les unités utilisées. L'attribut `IMAGEMAP` spécifie que l'image fait partie d'une image interactive. Finalement, `ALIGN` indique où placer l'image: `BLEEDLEFT`, `BLEEDRIGHT`, `CENTER`, `LEFT`, `RIGHT`, `JUSTIFY`. `BLEEDRIGHT` ou `BLEEDLEFT` aligne l'image avec la fenêtre de l'outil de navigation, `RIGHT` et `LEFT` avec les marges.

1.6.4. <APPLET> et </APPLET>

Le marquage **<APPLET>** et **</APPLET>** permet d'insérer une applet JAVA dans une page WEB. Voici la syntaxe complète du marquage :

```
<APPLET
  CODEBASE = codebaseURL
  ARCHIVE = archiveList
  CODE = appletFile
  ALT = alternateText
  NAME = appletInstanceName
  WIDTH = pixels  HEIGHT = pixels
  ALIGN = alignment
  VSPACE = pixels  HSPACE = pixels
>
  <PARAM NAME = appletAttribute1 VALUE = value>
  <PARAM NAME = appletAttribute2 VALUE = value>
</APPLET>
```

CODE, **WIDTH** et **HEIGHT** sont les attributs obligatoires. Décrivons à présent chacun des attributs.

L'attribut **CODEBASE** spécifie le lieu où se trouve l'applet à charger. Si cet attribut n'est pas spécifié, le navigateur utilise l'adresse du document en cours. **Archive** sert à spécifier les fichiers d'archives (fichiers JAR) à précharger, ceux-ci étant séparés par des virgules. L'attribut **CODE** spécifie le nom du fichier contenant le code compilé de l'applet.

On utilise **ALT** pour indiquer le texte à afficher au cas où l'applet ne pourrait pas être chargée (par exemple si le navigateur n'est pas compatible avec JAVA). **NAME** sert à donner un nom à une instance d'une applet permettant ainsi à des applets se trouvant sur la même page de se trouver et de communiquer. Les attributs **WIDTH** et **HEIGHT** spécifient la taille en pixels, respectivement la largeur et la hauteur, de la zone d'affichage de l'applet. **ALIGN** est utilisé pour spécifier l'alignement de l'applet, les valeurs possibles sont identiques à celles du marquage **IMG**. Les attributs **VSPACE** et **HSPACE** sont utilisés pour indiquer l'espace à réserver de chaque côté de l'applet, respectivement horizontalement et verticalement. Enfin, on utilise **<PARAM NAME = appletAttribute1 VALUE = value>** lorsqu'une applet nécessite des attributs pour s'exécuter.

Voici un exemple de l'utilisation de ce marquage :

```
<applet code="Une_Applet.class" width=100 height=140></applet>
```

Ce marquage signifie que le navigateur doit charger l'applet dont le code compilé est `Une_Applet.class` et réserver une taille de 100 pixels de large et de 140 pixels de haut.

Voici un autre exemple plus complet :

```
<applet codebase=" http://www.serveur.suffixe/sous_répertoire"
      code="AfficheTexte.class" width=400 height=75 VSPACE=20 HSPACE=20>
<param name="text" value="Coucou">
<hr>
Si votre navigateur supportait JAVA, alors vous devriez voir apparaître
Coucou
<hr>
</applet>
```

Ce marquage signifie que le navigateur doit charger l'applet dont le code compilé est `AfficheTexte.class`, se trouvant dans le répertoire `http://www.serveur.suffixe/sous_répertoire`, de l'afficher dans une zone de 400 pixels de large par 140 pixels de haut, en laissant autour un espacement de 20 pixels. Le navigateur doit aussi placer le texte "Coucou" comme paramètre de l'applet. Si le navigateur ne supporte pas JAVA, alors le texte "Si votre navigateur supportait JAVA, alors vous devriez voir apparaître Coucou" est affiché, sinon l'applet est chargée.

2. CONCLUSION

Le langage HTML permet la création de pages WEB attractives grâce à une syntaxe riche mais assez simple. Cette syntaxe, qui est en fait composée d'un ensemble de marques, permet de structurer un document, d'attribuer des styles au texte, d'utiliser des listes et de créer des tableaux. L'intérêt principal du langage est bien entendu qu'il est facilement possible de créer des liens hypermédias, qu'il s'agisse de liens vers de l'hypertexte, des images ou des pages WEB.

Toutefois, s'il permet d'afficher du texte et des images, il n'accepte pas les fichiers audio ni video sans l'aide des plug-ins du navigateur. De plus, il ne permet pas le chargement de programmes sur le WEB. C'est donc tout naturellement que ce manque a été comblé par la création et l'utilisation de programmes JAVA au sein de HTML. Cette "collaboration" permet au World Wide Web de devenir plus interactif et plus attrayant.

Deuxième partie :

Réalisation d'une application pour le WWW

Chapitre III :

L'application ChercheWEB

0. INTRODUCTION

Le présent chapitre traite de l'application ChercheWEB, application de recherche des liens se trouvant sur une page WEB. Tout d'abord, nous donnons l'énoncé (section 1). Ensuite, nous évoquons les problèmes posés et les solutions apportées (section 2). Puis, nous présentons l'application terminée¹ (section 3). Enfin, nous traitons des améliorations que l'on peut apporter à l'application ChercheWEB (section 4) et nous essayons de voir quelles utilisations on peut faire de l'application améliorée (section 5). Nous terminerons le chapitre par une conclusion (section 6).

1. ENONCE

Le Département des Communications de L'Université du Québec à Montréal, et plus particulièrement Mr Louis-Claude Paquin, était intéressé par la programmation JAVA. Il souhaitait faire quelques essais afin de savoir ce qui était possible. Il a donc demandé de réaliser un programme JAVA qui satisferait aux spécifications suivantes :

1. l'utilisateur doit entrer une adresse WEB du type `http://www.nom.suffixe` ;
2. le programme se connecte à l'adresse indiquée ;
3. le programme renvoie la liste des liens se trouvant dans cette page.

¹ Les spécifications du programme ainsi que les copies d'écran se trouvent en annexe.

Une fois cela fait, certaines améliorations étaient possibles :

1. permettre à l'utilisateur d'enregistrer le résultat de sa recherche ;
2. permettre à l'utilisateur d'ouvrir des fichiers textes ;
3. faire une recherche récursive à partir des liens trouvés dans la page ;
4. afficher les résultats sous forme graphique.

2. PROBLEMES POSES ET SOLUTIONS APPORTEES

Le premier problème posé, et le plus important, fut la sécurité des informations et la distinction qui en découle entre les deux types de programme JAVA : les applets et les applications. Une applet ne peut fonctionner que si elle est incluse dans une page WEB, elle dépend donc d'un browser Internet (Netscape ou Microsoft Explorer, par exemple) . Une application est, elle, indépendante du browser. Au départ, le programme a été réalisé sous forme d'applet, mais dans ce cas, pour des raisons de sécurité, le browser ne permet pas que l'on se connecte sur un site pour en retirer des informations.

Il a donc fallu transformer l'applet en application car les applications n'imposent pas de telles restrictions. Toutefois, la transformation d'une applet en application ne se fait pas aussi facilement que décrit dans les manuels. De plus, tous les livres actuels concernant JAVA sont écrits pour des applets et bien peu traitent des applications et cela peut poser des problèmes car ces deux types de programmes ne prennent pas en charge certaines fonctions de la même façon (par exemple pour le traitement d'images et de sons).

Le deuxième problème posé fut celui du blocage de l'application. Lorsque le programme se connectait à la page WEB, cela pouvait prendre beaucoup de temps et bloquer ainsi tous les processus. Il devait donc être possible d'interrompre la recherche. Pour ce faire, la solution était d'utiliser des threads (un thread est un processus qui peut s'exécuter en parallèle avec d'autres processus.)

Le troisième problème fut celui de l'interface. En effet, la version du JAVA Development kit utilisée à l'époque (JDK 1.0.2) est limitée au niveau des objets de l'interface, elle ne permet que les boutons radios, les champs d'édition, les listes de sélection, les boutons de commande, les boîtes à cocher et les menus. De plus, il faut programmer ces objets. Donc la plupart du code du programme est du code de l'interface.

3. L'APPLICATION TERMINEE : CHERCHEWEB

L'application réalisée satisfait aux trois conditions de l'énoncé ainsi qu'aux deux premières demandes d'amélioration. L'application ChercheWEB est composée de 3 menus :

1. un menu "Fichier" ;
2. un menu "Historique" ;
3. un menu "?".

3.1. LE MENU "FICHIER"

Le menu "Fichier" est composé de 4 sous menus.

Le sous menu "Ouvrir..." affiche une boîte de dialogue pour choisir le fichier texte à ouvrir et en affiche le contenu. Le sous menu "Enregistrer..." permet d'enregistrer le résultat de la dernière recherche. Il affiche une boîte de dialogue pour saisir le nom du fichier dans lequel enregistrer ce résultat. En plus du résultat de la recherche, l'utilisateur trouvera dans le fichier la date de la recherche et l'adresse de la page où la recherche a été effectuée. Le sous menu "Quitter..." permet de sortir du programme. Une boîte de dialogue est affichée pour demander à l'utilisateur s'il est certain de vouloir quitter. L'utilisateur peut alors choisir "Oui" s'il désire effectivement sortir de l'application ou "Non" dans le cas contraire. Le sous-menu "Rechercher" affiche la boîte de dialogue "Rechercher..." qui est le coeur de l'application ChercheWEB. Elle permet de parcourir une page WEB et d'afficher les liens contenus dans cette page WEB.

Voyons comment effectuer une recherche.

Avant tout, l'utilisateur doit remplir la zone intitulée : "Veuillez introduire l'URL à parcourir". Ensuite, il doit cliquer sur le bouton "Rechercher" ou appuyer sur la touche [Entrée]. Si l'URL est valide, le programme va s'y connecter et rechercher les liens contenus. Si l'URL n'est pas valide, l'utilisateur recevra un message d'erreur. L'utilisateur peut arrêter la procédure en appuyant sur le bouton "Stop".

Une fois la recherche effectuée, les liens contenus dans l'URL sont affichés dans la zone : "Voici ce que contient cet URL". Dès lors, il est possible de recommencer la recherche à partir d'un des liens trouvés en sélectionnant le lien et en appuyant sur le bouton "Chercher".

Le bouton "Effacer" permet d'effacer la zone "Veuillez introduire l'URL à parcourir". Le bouton "Quitter" permet de sortir de la recherche.

La figure III.1 représente la boîte de dialogue "Rechercher..."

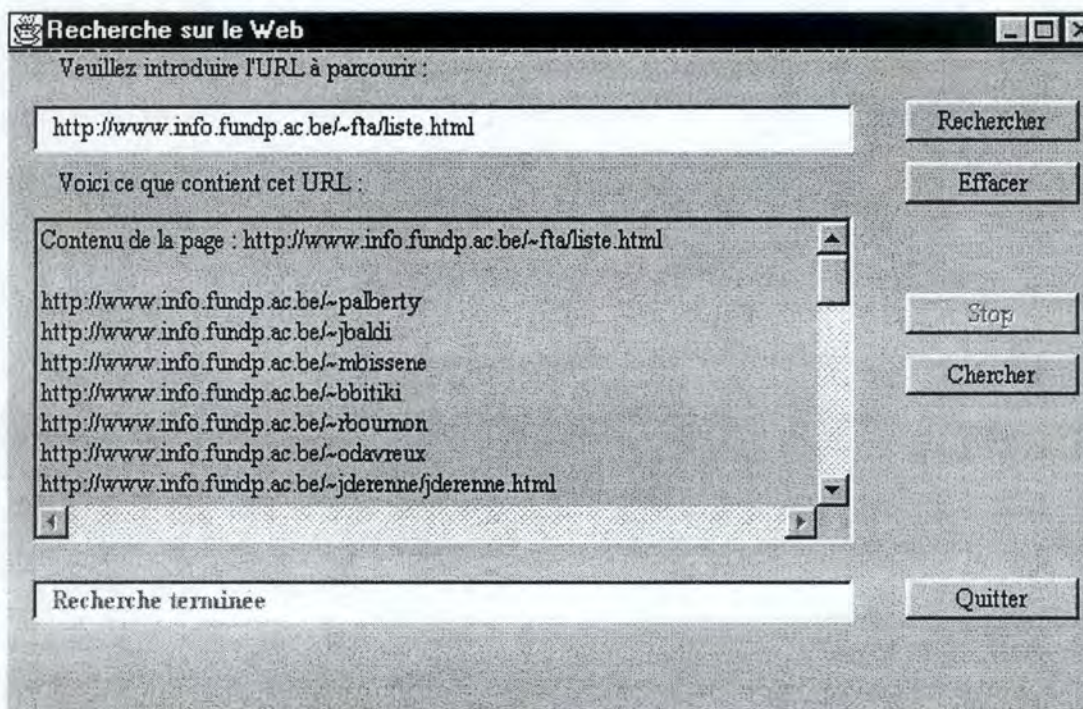


Figure III.1 : La boîte de dialogue "Rechercher..."

Explication de la recherche

Avant toute chose, un thread est créé afin d'accélérer la recherche et de pouvoir la stopper quand on le désire. L'application se connecte alors à la page WEB dont l'utilisateur a donné l'adresse (si cette adresse est valide. Sinon, un message d'erreur est renvoyé). L'application crée alors un nouveau flux de données (InputStream dans le langage JAVA), afin de pouvoir lire le contenu de la page WEB. Une fois cela fait, la recherche commence au sein de cette page WEB : tant qu'on n'est pas arrivé à la fin du fichier HTML, on vérifie ligne par ligne la présence de l'identificateur de lien `href="http://`. Si on trouve un tel identificateur, on copie ce qui vient après dans un tampon (ex : si on trouve `href="http://www.nom.suffixe/page.html"`, on place `www.nom.suffixe/page.html` dans le tampon). On considère que le lien est trouvé quand on atteint des guillemets. Lorsque l'on est à la fin du fichier, on ferme le flux de données, on copie le contenu du tampon dans la zone prévue pour l'affichage des liens.

3.2. LE MENU "HISTORIQUE"

Le menu "Historique" est composé du sous menu "Historique...". Celui-ci affiche une boîte de dialogue qui permet de voir les pages parcourues au cours de l'ensemble des recherches. Les adresses des pages parcourues sont affichées les unes en dessous des autres dans l'ordre chronologique du parcours.

3.3. LE MENU "?"

Le menu "?" est composé du sous menu "A propos de..." qui affiche la boîte d'informations sur le programme ChercheWEB.



4. AMELIORATIONS DE L'APPLICATION

En apportant quelques améliorations, il serait possible de rendre l'application plus complète afin de pouvoir l'utiliser dans des contextes d'aide à la navigation et d'analyse sémantique des pages WEB.

Au nombre des améliorations que l'on pourrait apporter à l'application, nous pouvons citer la possibilité d'une recherche récursive, l'affichage graphique des résultats, l'ouverture du document avec un navigateur, le plan de la page et la recherche sur critère.

La **recherche récursive** permettrait de ne plus se contenter d'un niveau de recherche mais de parcourir plusieurs niveaux dans la structure du WEB. Afin d'éviter que l'application ne se termine jamais, il serait nécessaire d'indiquer quand arrêter la recherche (par exemple quand on a trouvé x liens, quand on a atteint x niveaux, quand on change de serveur ...). Le contenu des différentes recherches pourrait se trouver dans différentes fenêtres permettant ainsi de visualiser le contenu de plusieurs pages en même temps.

Afin de visualiser plus facilement le contenu des recherches, un **affichage graphique** des résultats est utile : il permettrait d'améliorer la présentation de la structure existant entre les pages. Il serait intéressant d'ouvrir une nouvelle fenêtre dans laquelle on indiquerait les différentes pages parcourues reliées par des fils, qui pourraient être de couleurs, de formes différentes selon la proximité. Une telle représentation permettrait de découvrir l'existence de boucles entre pages (pages se référant entre elles), de pages finales (pages ne comportant pas de références vers d'autres pages)... Cette amélioration est tout à fait réalisable car JAVA offre des objets en 2 dimensions (carrés, cercles ...) et en 3 dimensions (cubes, sphères...) pour créer cet affichage graphique. Il suffirait alors de créer un tel objet pour chaque lien trouvé. Une présentation possible de cet affichage serait celle utilisée par les gestionnaires de fichiers de Windows : on utiliserait un objet pour une page et il serait possible, en cliquant sur cet objet, d'afficher la liste des pages référencées. Cette présentation permettrait à l'utilisateur de développer ou de réduire le graphe des pages.

Tout comme avec Windows, un symbole (par exemple ) à côté de l'objet indiquerait qu'il est possible de développer le graphe alors qu'un autre symbole (par exemple ) signifierait qu'il n'est plus possible de développer le graphe.

L'affichage des pages, sous forme graphique, ne donne pas un aperçu du contenu de la page. Pour que l'utilisateur puisse prendre connaissance des informations se trouvant sur la page, il serait intéressant d'ajouter à l'application la possibilité d'**ouverture du document avec un navigateur**. Pour ce faire, il suffit d'ajouter un bouton de commande qui déclencherait le lancement du navigateur avec l'adresse de la page WEB référencée. Une autre façon de faire serait de cliquer directement sur l'objet représentant la page référencée pour l'ouvrir avec le navigateur. En effet, le langage JAVA permet d'exécuter une commande sur le système d'exploitation, en l'occurrence l'appel au navigateur.

Afin d'aider l'utilisateur dans sa recherche, sans qu'il doive ouvrir le document avec le navigateur, l'application pourrait fournir un **plan de la page** référencée par le lien, par exemple en recueillant l'ensemble des titres ou l'ensemble des images.

Une dernière amélioration au programme pourrait être la **recherche sur critère**. Avant de lancer sa recherche, l'utilisateur indiquerait un critère concernant le type de document qu'il souhaite ou alors un ensemble de mots clés. La recherche ne se ferait alors plus uniquement sur base de l'identificateur `href="http://` mais aussi selon le critère imposé par l'utilisateur.

5. UTILISATIONS POSSIBLES

Les utilisations possibles de l'application améliorée sont la recherche exhaustive des liens d'un site, l'élaboration de la structure d'un site, la classification selon un mot clé et l'analyse sémantique d'une page. Donnons quelques mots d'explication concernant ces utilisations possibles.

Il est parfois utile de connaître la liste des liens qui se trouvent sur un site, ne fut-ce que pour pouvoir les mettre à jour ou pour avoir une idée des informations que contient le site. Avec l'application améliorée, il est possible d'avoir la liste de tous ces liens en menant une recherche et en posant comme critère d'arrêt de la recherche le moment où l'on change de site. De plus, cette **recherche exhaustive** permet de tracer la **structure d'un site** en indiquant les liens internes, les liens externes, les références dans un même document. En utilisant l'affichage graphique, on peut visualiser cette structure.

L'application pourrait également être une aide aux moteurs de recherche actuels. En utilisant un mot clé comme critère de recherche, il est possible d'établir une **classification** des pages par sujet. De même, si le site n'a pas de table des matières, de plan du site ou d'utilitaire de recherche, l'application permet rapidement d'atteindre la ou les pages recherchées.

Si l'utilisateur demande d'afficher non seulement la liste des liens mais également les titres contenus dans une page, il est possible d'en faire une analyse sémantique. Ainsi, sans avoir besoin de lancer l'ouverture de la page avec un navigateur, l'utilisateur peut se faire une idée des sujets qui y sont développés.

6. CONCLUSION

L'application ChercheWEB est toute simple : elle se connecte à une page WEB et en dégage les liens. Cependant, si on lui apporte quelques modifications, il est possible d'étendre son champ d'utilisation. Ainsi, elle peut devenir une aide à la navigation, un outil de structuration des pages WEB et même un analyseur sémantique. Il faut cependant remarquer que l'analyse sémantique que l'on peut mener actuellement est restreinte. Ceci est dû au fait que le langage HTML, utilisé pour créer des pages WEB, contient peu (voire pas) de champs (ou de marques) permettant de laisser un résumé de la page ou encore la liste des sujets. Il serait utile de les ajouter afin de permettre une analyse plus facile et plus automatique. Pour cela, il est nécessaire de bien comprendre les différents types de liens existants, les différentes façons de les représenter ainsi que de fournir différents moyens permettant de mieux comprendre le contenu des pages référencées par ces liens. C'est ce que nous allons développer dans la troisième partie de ce mémoire.

Troisième Partie :

Les Liens sur le World Wide Web

Chapitre IV : Les liens

0. INTRODUCTION

Dans le présent chapitre, nous traitons des liens hypermédias. Dans le but de nous rendre compte de ce qui existe actuellement au niveau des liens sur le World Wide WEB, nous traçons tout d'abord la typologie des liens hypermédias (section 1). Nous classifions les liens présents sur le WEB et nous leur appliquons cette typologie des liens (section 2) et nous donnons quelques exemples de représentation de ces liens (section 3). Ensuite, nous voyons comment retirer des informations de l'examen d'un lien (section 4) et nous examinerons quelques pages WEB (section 5). A partir de cet examen, nous exposons les moyens de présentation permettant d'apporter un complément d'information sur les liens (section 6) et nous proposons quelques améliorations possibles facilitant la navigation de l'utilisateur (section 7). Enfin, nous terminons par une conclusion (section 8).

1. TYPOLOGIE DES LIENS HYPERMEDIAS

Sur base des différents types de liens¹ existant à l'heure actuelle, nous pouvons établir une classification en fonction de la nature des liens, de la possibilité de leur identification automatique, de la caractérisation de leur destination et de leurs caractéristiques physiques.

¹ Basé sur le chapitre 2 de [Serpe].

1.1. NATURE

On peut distinguer les liens en fonction de leur nature :

- les liens **structurels** ont pour fonction de faciliter la lecture en présentant la structure existant entre des documents qu'ils mettent en relation ;
- les liens **sémantiques** ont pour fonction d'apporter des informations sémantiques concernant la relation existant entre des documents qu'ils mettent en relation.

1.2. POSSIBILITE D'IDENTIFICATION AUTOMATIQUE

Une autre façon de distinguer les liens est la possibilité de les identifier plus ou moins automatiquement. On peut avoir des liens **manuels** (qui nécessitent l'intervention humaine pour leur identification), des liens **semi-automatiques** et des liens **automatiques**.

1.3. CARACTERISTIQUES DE DESTINATION

Les caractéristiques de destination sont un autre moyen de classifier les liens. Ces caractéristiques sont :

- la **granularité** qui précise si l'on a affaire à un document ou à une partie de document ;
- le **format des données** qui peut être de type texte, audio, vidéo...
- la **taille et l'emplacement** qui permettent d'évaluer le coût de déplacement d'un lien à un autre ;
- l'**accès** au document qui peut être privé ou public ;
- le **coût** qui indique si le document est gratuit ou payant ;
- la **date de mise à jour** qui permet de savoir si une page comporte de nouvelles informations.

1.4. CARACTERISTIQUES PHYSIQUES

Les caractéristiques physiques suivantes permettent d'établir une autre classification :

- le **degré de détermination** distingue les liens selon que leurs extrémités sont déterminées (liens déclarés), peu déterminées (liens dynamiques) ou très peu déterminées (liens extensibles).
- la **cardinalité** distingue les liens en fonction du nombre de documents qu'ils référencent. On a ainsi les liens simples (qui relient un document source à un document destination) et les liens multiples (qui relient un document source à plusieurs documents de destination).
- l'**orientation** détermine le sens de parcours des liens. Ils peuvent être parcourus dans un seul sens (liens unidirectionnels) ou dans les deux sens (liens bidirectionnels).
- la **visibilité** se base sur l'ancrage des liens à leur source. Le lien est explicite si l'ancrage est visible et implicite dans le cas contraire.

2. LIENS EXISTANT SUR LE WEB

Dans cette section, nous proposons une classification simple des liens selon trois axes (la présentation des liens, leur localisation et le type de document qu'ils référencent) :

- la **présentation** est la façon dont l'existence du lien est signalée à l'utilisateur. Actuellement, on trouve des liens dans le texte (le lien vers un autre document est une partie du texte), des liens sur image (le lien est une image sur laquelle il faut cliquer), et des "liens dans une carte" (une carte est une image divisée en zones qui peuvent référencer des documents différents).
- la **localisation** permet de distinguer les liens en fonction de la distance entre la page contenant l'ancrage du lien et le document référencé. On a ainsi des liens internes (sur le même site, mais pas nécessairement sur la même page), des liens externes (sur un autre site). Pour ces deux types de liens, on peut également avoir des liens de référence (sur une partie de document).
- les différents types de documents peuvent être du texte, des images, du son, des séquences vidéos ou des fichiers (généralement compressés).

Reprenons à présent la typologie présentée dans la section précédente et voyons ce qui est applicable aux liens présents sur le WEB.

En ce qui concerne la nature des liens, on a affaire aussi bien à des liens structurels, qui précisent la structure du document (ex : liens permettant de remonter au début de la page, à la page principale du site...), qu'à des liens sémantiques qui fournissent des informations sur des sujets exprimés dans la page. Cependant, on remarque que les liens sémantiques sont plus nombreux que les liens structurels.

Sur le WEB, l'identification automatique des liens peut se faire en examinant l'adresse. Comme nous le verrons dans la section 4, il est possible de retirer certaines informations de cet examen. Toutefois, celui-ci ne permet pas une identification automatique, tout au plus, pouvons-nous avoir une identification semi-automatique.

Certaines caractéristiques de destination peuvent être découvertes : grâce à un examen du lien, il est possible de connaître la granularité (dans la plupart des cas, on trouve des liens vers des documents, mais on trouve aussi des liens vers des parties de documents). Le format des données est connu par le suffixe du fichier (html, gif...), l'emplacement est précisé par le nom et le domaine du serveur. Par contre, la taille du document, les conditions d'accès et le coût sont rarement connus.

Enfin, pour ce qui est des caractéristiques physiques, nous nous trouvons face à des liens déclarés (leurs extrémités sont connues), simples (les liens référencent un seul document), explicites (leur ancrage est généralement visible) et unidirectionnels (au niveau des liens, une fois que l'on a accédé à un document, il n'est pas possible de revenir à leur document source. A ce propos, remarquons que si l'on prend pour cadre, non plus simplement un lien, mais un document entier, il est possible d'avoir le caractère bidirectionnel en insérant un lien vers la source dans le document de destination).

3. REPRESENTATION DE LA STRUCTURE DES LIENS

Afin d'aider l'utilisateur à naviguer au sein des pages WEB, une représentation de la structure des liens est parfois utile, surtout lorsque cette dernière est complexe. Une bonne représentation doit limiter les phénomènes de "surcharge cognitive" et de "désorientation", et constituer une aide et non pas un obstacle à la navigation. La désorientation se produit lorsque l'utilisateur ne se retrouve plus dans la structure des pages parcourues. Il ne sait ni comment il est arrivé sur la page actuelle ni quel lien choisir pour trouver rapidement et efficacement l'information dont il a besoin. Tous ces facteurs combinés entraînent la surcharge cognitive.

Nous allons présenter quelques exemples de représentations¹ existant actuellement. Elles sont principalement axées autour des cartes en deux dimensions (section 3.1) et des métaphores, généralement en trois dimensions (section 3.2).

3.1. LES CARTES EN DEUX DIMENSIONS

Généralement, les cartes en deux dimensions, utilisés pour représenter la structure du WEB, sont apparentés à des graphes au sens large : ils mettent en relation les noms des pages (noeuds du graphe) au moyen de fils (arcs du graphe). Ainsi, on place un fil entre une page contenant l'ancrage d'un lien et la page vers laquelle pointe ce lien.

Une telle représentation est facile à réaliser et à visualiser quand la structure est simple, mais dès que celle-ci se complexifie, la représentation par "graphe" perd ces deux qualités.

¹ Basé sur [Rouard]

Afin d'aider l'utilisateur, des représentations telles que le Navigational View Builder¹ (cfr. Figure V.1) utilisent des icônes pour les noeuds (une icône par type de document) et des styles de liens différents, selon la proximité, la taille et le sujet du document de destination. On remarque cependant que cette représentation est surchargée et pratiquement illisible pour les cas où la structure est complexe.

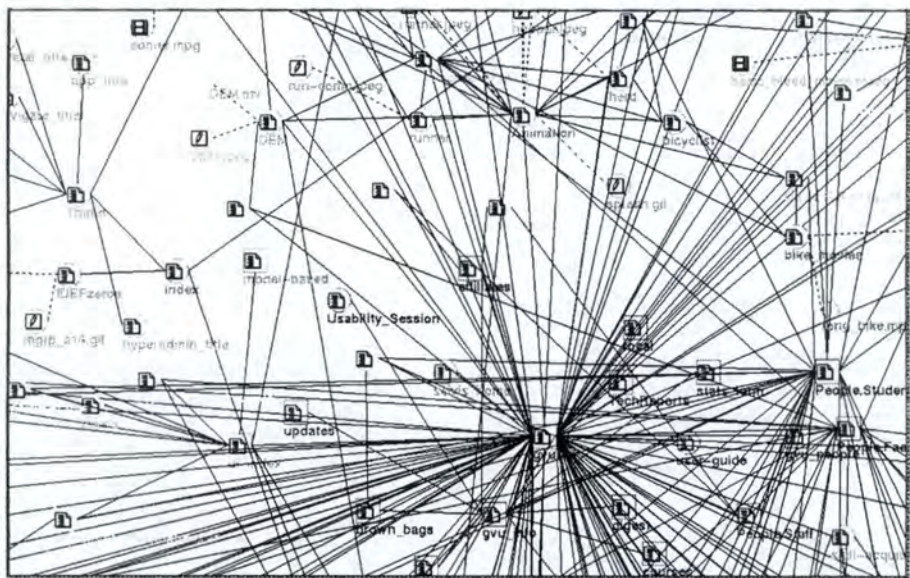


Figure V.1 : Une des premières représentations par Navigational View Builder

Les icônes utilisées pour représenter les types de documents pourraient ressembler à celles illustrées à la Figure V.2 qui sont celle utilisées par le système *World Wide Web Ressource Discovery*² :

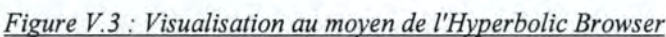


Figure V.2 : Icônes représentant les types de document

¹ Tiré de [Mukherjea]

² Tiré de [Yuwono]

L'Hyperbolic Browser (cfr. Figure V.3), une adaptation de la représentation fish-eye view, offre la possibilité d'afficher 1000 noeuds dont 50 près du centre et de déplacer ce centre d'intérêt. Cette technique convient très bien lorsqu'il s'agit d'une hiérarchie, mais en ce qui concerne le WEB, on a rarement affaire à une hiérarchie...



Chapitre IV : Les Liens

Au vu de ces exemples, on s'aperçoit de la difficulté de représentation des liens : lorsque les liens sont nombreux et qu'il n'y a pas de hiérarchie, le graphe est embrouillé et est incompréhensible pour l'utilisateur. Dès lors, les concepteurs de représentation ont pensé à utiliser des métaphores pour visualiser la structure du WEB.

3.2. LES METAPHORES

Les métaphores ont pour but de faciliter la compréhension de la structure du WEB au moyen d'objets connus par l'utilisateur. Elles sont généralement basées sur les objets en trois dimensions. Les objets 3D offrent plus de possibilités de reconnaissance et de classification que les graphes : il est possible de créer des objets de formes et de tailles différentes, de les faire tourner dans l'espace...

La représentation utilisée pour le projet *Measuring the WEB*¹ (cfr. Figure V.4) nous donne un exemple de représentation au moyen des objets 3D. Pour chaque site, la hauteur des objets varie en fonction du nombre de liens vers ce site, la taille des sphères en fonction du nombre de liens de ce site vers l'extérieur, le diamètre du socle indique la taille du site et le domaine est représenté par sa couleur. Enfin, la connectivité est représentée par la répartition des objets dans l'espace les uns par rapport aux autres.

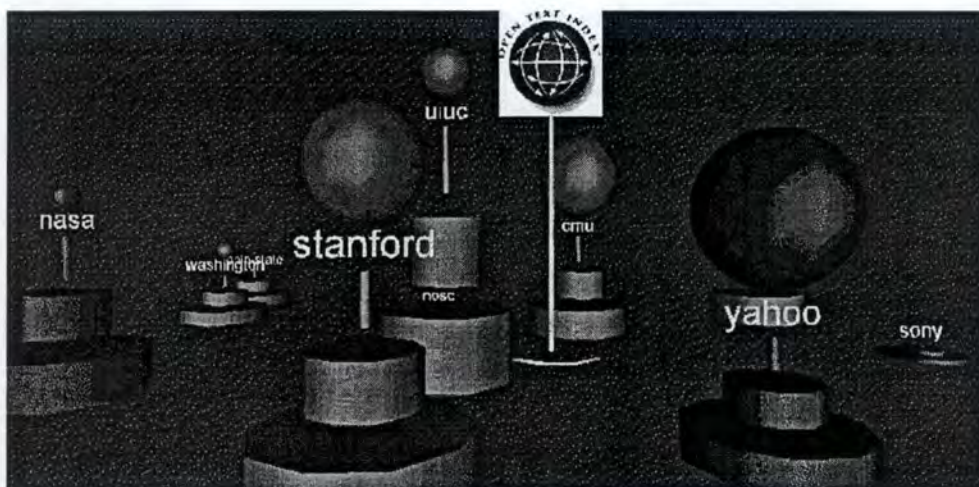


Figure V.4 : Représentation au moyen d'objets 3D

¹ Tiré de [Bray]

Une autre métaphore utilisée pour représenter la structure du WEB est celle du livre reprise par le WebBook¹ (cfr. Figure V.5). Les pages reprises dans ce livre peuvent être composées à partir des liens se trouvant sur une page, à partir des résultats d'une recherche ou à partir de signets (bookmarks) définis par l'utilisateur ; elles peuvent également être groupées par sujet.

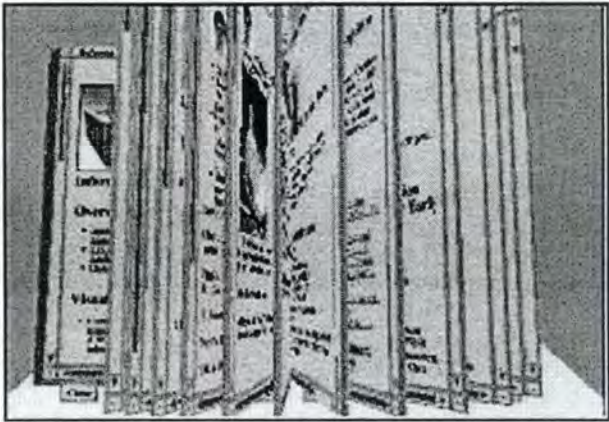


Figure V.5 : Représentation WEBBook

Le WEBBook s'insère dans une représentation plus complète, appelée WEBForager (cfr. Figure V.6), et composée de deux niveaux supérieurs : le Desk, qui peut contenir plusieurs livres et/ou pages individuelles, et le BookCase, représenté par une étagère contenant l'ensemble des livres.

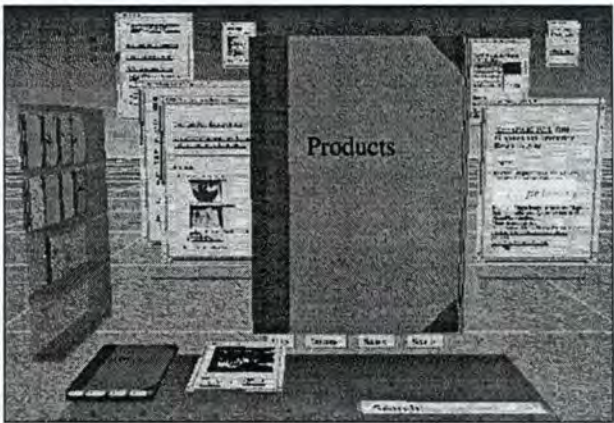


Figure V.6 : Représentation du BookForager

¹ Tiré de [Card]

4. EXAMEN D'UNE ADRESSE WEB

Après avoir passé en revue les différents types de liens et quelques unes de leurs représentations, voyons à présent quelles informations nous pouvons tirer de l'analyse de l'adresse pointée par un lien. Ainsi, on peut examiner une adresse en fonction du préfixe de l'adresse, du serveur, et du suffixe de la page.

4.1. PRÉFIXE DE L'ADRESSE

Nous appelons préfixe de l'adresse ce qui se trouve en début de cette adresse, avant les deux points et les barres divisé (://), et qui sert généralement à identifier le protocole utilisé. La liste ci-dessous reprend les préfixes utilisés le plus fréquemment :

- **http :** permet l'accès à un serveur HyperText Transport Protocol, de transport des fichiers hypertextes ;
ex : `http://www.belnet.be`
- **ftp :** permet l'accès à un serveur File Transfer Protocol, de transfert de fichiers ;
ex : `ftp://ftp.belnet.be`
- **gopher :** permet l'accès à un serveur Gopher indiquant les éléments disponibles sur un serveur ;
ex : `gopher://gopher.belnet.be`
- **telnet :** permet une connexion à distance sur un serveur ;
ex : `telnet://belnet.be`
- **news :** permet l'accès à un serveur de groupes de discussion (newsgroup) ;
ex : `news://hermes.fundp.ac.be/`
- **mailto :** permet l'envoi d'un mail ;
ex : `mailto:www@info.fundp.ac.be`
- **irc :** permet l'accès à un serveur Internet Relay Chat de discussion multi-utilisateurs ;
ex : `irc://belnet.be`
- **file :** permet l'accès à un fichier sur un disque local ;
ex : `file:///C:/Netscape/bookmark.htm`

4.2. SERVEUR

Il est possible de retirer des informations en examinant le nom et le suffixe du serveur dans l'adresse. Le nom, bien souvent, est suffisamment explicite pour indiquer le sujet général du site. Le suffixe nous permet de connaître le pays dont est originaire le serveur ou d'en connaître le domaine.

- nom de pays : le nom du pays est indiqué par un suffixe de deux lettres.
ex : "be" pour Belgique, "ca" pour Canada, "uk" pour Royaume Uni.
- domaine de serveur
 - com : serveur commercial (ex : `www.microsoft.com`) ;
 - mil : serveur militaire aux USA (ex : `www.navy.mil`) ;
 - edu : serveur éducatif, universitaire aux USA (ex : `www.cwru.edu`) ;
 - org : serveur d'une organisation (ex : `www.unesco.org`) ;
 - gov : serveur gouvernemental (ex : `www.whitehouse.gov`) ;
 - net : serveur faisant partie d'un réseau (ex : `www.imsWEB.net`) ;

4.3. SUFFIXE DE LA PAGE

Le suffixe de la page nous permet de connaître le type de document auquel on va accéder. Voici quelques-uns des suffixes de page le plus fréquemment rencontrés :

- html ou htm: document HyperText Markup Langage ;
- txt : document de type texte ;
- au : fichier audio au format *au* ;
- gif : image au format *gif* ;
- jpg : image au format *jpeg* ;
- avi : fichier vidéo au format *avi*, Video for Windows ;
- zip : fichier compressé au format *zip* ;

5. EXAMEN DE PAGES WEB

Dans le présent chapitre, nous examinons les pages WEB¹ de certains sites appartenant aux secteurs des banques (section 5.1), de l'immobilier (section 5.2) et des fournisseurs d'accès à Internet (section 5.3). Nous essayons de voir quels moyens ont mis en œuvre ceux qui ont réalisé ces sites pour apporter un complément d'information sur les liens. Au sein de chacun des secteurs, nous voyons les thèmes communs et les différences entre les sites. Pour terminer cet examen, nous présentons les observations communes aux trois secteurs (section 5.4).

5.1. SECTEUR DES BANQUES

Pour cet examen des pages WEB dans le secteur bancaire, nous nous sommes basés sur l'analyse des sites de la [Bacob], la [BBL], la [CGER], le [Crédit] communal et la Générale de Banque [Gbank].

En parcourant ces sites, on s'aperçoit que, généralement, nous trouvons des liens sur images, dans le texte, et quelques liens de référence. La plupart sont des liens sémantiques, ils pointent vers des documents apportant de l'information, et internes, ils référencent des pages sur le même serveur. Il y a aussi quelques liens structurels, par exemple ceux qui permettent de revenir à la page d'accueil.

On note également l'utilisation de frames (qui permettent de découper l'écran du navigateur en plusieurs documents HTML) pour la Bacob, la CGER et le Crédit Communal, une frame où se trouvant à la page d'accueil donne des informations sur les grandes divisions du site. La BBL utilise des infobulles mais simplement pour y inscrire le même mot que celui qui compose le lien. La Générale de Banque, quant à elle, n'apporte aucune information concernant ses liens. Bien souvent, tout ce dont on dispose pour se faire une idée des informations contenues dans la page est l'image qui compose le lien.

¹ Les copies d'écran des pages les plus représentatives se trouvent en annexe.

La Figure V.7 représente une partie de la page WEB utilisant les frames pour apporter un complément d'information. Lorsque le curseur de souris passe sur Jcontact dans la frame de droite, le texte "Les avantages de la Jcard. Recherchez dans notre dictionnaire financier" s'affiche dans la frame du haut.

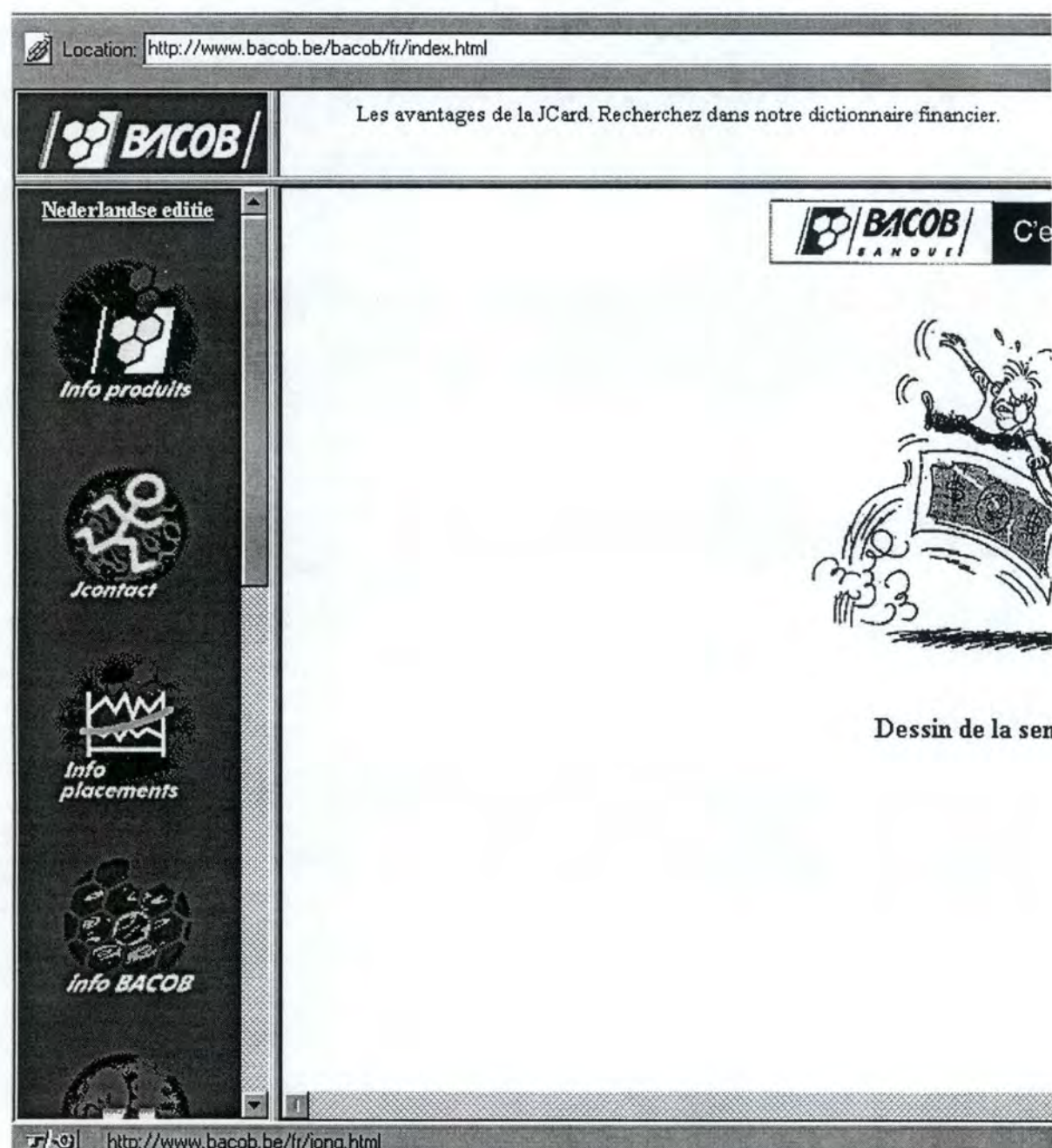


Figure V.7 : Page WEB comportant des frames

Enfin, remarquons qu'en plus des frames, le site de la Bacob utilise une fenêtre de navigation écrite en JAVA (cfr. Figure V.8), celle-ci est affichée en permanence et permet de se déplacer en tout point du site de manière rapide et efficace. Cela nous semble un bon exemple d'aide à la navigation.

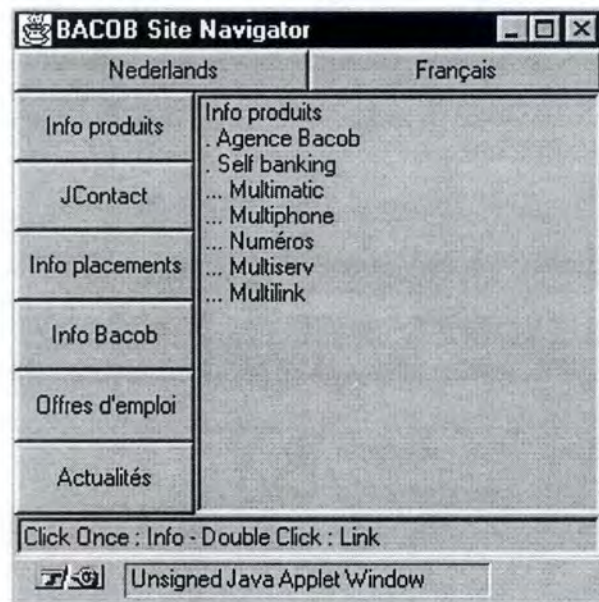


Figure V.8 : Fenêtre de navigation de la Bacob

Thèmes repris par tous les sites bancaires :

- informations bancaires en général (en particulier l'Euro) ;
- informations sur la société (localisation, chiffre d'affaires...) ;
- produits de la société (placements, assurances, ...), localisation des agences, voyages ;
- offres d'emploi ;
- partenaires ou sociétés à l'étranger ;
- recherche sur le site ;
- liens divers (JudoNet, nouvelle orthographe flamande...).

On peut remarquer que le secteur bancaire est relativement fermé sur lui-même : il y a très peu de liens externes. C'est également un secteur dont les thèmes sont assez déterminés. On regrette cependant que les termes ne soient pas standardisés (ex : les agences se trouvent dans la rubrique "produits" à la Bacob, dans la rubrique "privé" à la BBL, dans la rubrique "ma banque" à la CGER), ce qui peut désorienter l'utilisateur au cours de sa navigation. De même, certaines banques font des efforts pour la guidance (au moyen de frames, d'infobulles...) alors que d'autres laissent l'utilisateur se débrouiller.

5.2. SECTEUR DE L'IMMOBILIER

Pour cet examen des pages WEB dans le secteur de l'immobilier, nous nous sommes basés sur l'analyse des sites WEB de [Abelcom], [Cyberimmo], [Immobel], WEB de l'[Immobilier] et [Maisons] de la Côte.

Comme dans le secteur des banques, on trouve beaucoup de liens dans le texte et sur image, mais on trouve également des liens dans une carte, ce qui facilite la recherche dans un domaine où la géographie est fort présente. Les liens sont pour la plupart des liens sémantiques et internes.

Le site de la société Abelcom, le plus complet des sites examinés, contient des liens vers des sites externes (informations touristiques, culturelles, sportives, météorologiques, juridiques...) et utilise des frames pour réduire le phénomène de désorientation.

Thèmes repris par tous les sites immobiliers :

- informations sur la firme d'immobilier ;
- promoteurs et agences immobilières partenaires ;
- annonces immobilières (achat, vente, location) ;
- recherche selon critères ;
- liens divers (moteurs de recherche, placer une annonce...).

Le secteur de l'immobilier est encore peu présent sur le WEB. On sent que les sociétés immobilières ne sont pas très au fait des moyens existants dans la présentation de pages WEB. Dès lors, les sites sont très peu structurés et utilisent très peu les frames et les infobulles. De plus, comme dans le secteur bancaire, les informations sont présentes sur le site, mais on ne sait pas très bien où aller les chercher. Il serait donc, une fois de plus, intéressant de standardiser les dénominations.

5.3. SECTEUR DES FOURNISSEURS D'ACCES A INTERNET

Pour cet examen des pages WEB dans le secteur des fournisseurs d'accès à Internet, nous nous sommes basés sur l'analyse des sites WEB de [Arcadis], [Gate71], [Interpac], [Ping] et [Skynet].

Avec le secteur des fournisseurs d'accès, on trouve beaucoup plus de liens externes que dans les deux autres secteurs. Et pour cause, les pages de ces fournisseurs doivent être un point de départ pour les nouveaux utilisateurs d'Internet. Pour présenter ces liens, on trouve aussi bien des liens dans le texte, des liens sur images que des liens sur cartes. L'utilisation des frames est fréquente et peut permettre de structurer le site. Mais on manque manifestement d'informations sur les liens, ce qui peut prolonger les recherches sur le site.

Thèmes repris par tous les sites immobiliers :

- informations sur la société, points de présence ;
- types et conditions d'abonnement ;
- logiciels fournis et aide ;
- sites hébergés et pages personnelles ;
- sites intéressants pour commencer la navigation (moteurs de recherche, journaux...) ;
- liens divers (publicités, Mapi...).

Le secteur des fournisseurs d'accès se doit, bien entendu, d'être à la pointe du progrès en ce qui concerne Internet. Et de fait, nous remarquons que toutes les technologies actuelles sont présentes (utilisation de frames, d'infobulles, d'images, de sons, d'applets JAVA...). Toutefois, nous pouvons nous demander si cela n'est pas un obstacle à la navigation des utilisateurs novices (qui se demandent ce qu'est un cookie, ou un plug-in, cfr Figure V.9). Alors que les thèmes sont facilement identifiables et permettent une standardisation de la présentation, la structure des informations est fort différente d'un site à l'autre.

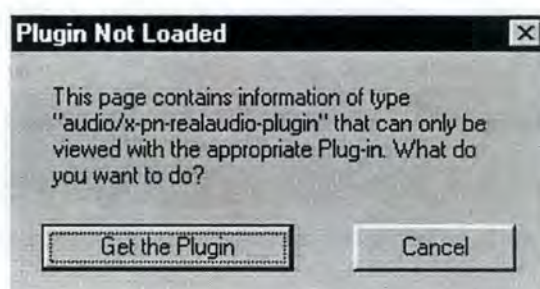


Figure V.9 : Exemple de demande de plug-in

5.4. OBSERVATIONS COMMUNES AUX TROIS SECTEURS

Si les trois secteurs examinés, le secteur des banques, le secteur de l'immobilier et le secteur des fournisseurs d'accès, véhiculent des informations très différentes, nous remarquons certaines similitudes en ce qui concerne leur passage sur le World Wide Web. Les frames, qui pourraient aider l'utilisateur dans sa navigation, sont peu ou mal utilisées et ne permettent pas de bien se situer dans la structure des sites. Au sein des secteurs, il y a un manque évident de standardisation : les termes désignant les mêmes sujets sont différents (ex : offres d'emploi, postes vacants...) et les pages contenant ces sujets sont accessibles par des liens peu reconnaissables.

L'analyse des trois secteurs nous permet de dégager des thèmes communs :

- informations sur la société ;
- informations sur les produits ;
- recherche ;
- liens divers.

Comme les secteurs examinés sont assez différents, nous pouvons supposer que ces thèmes sont repris dans tous les secteurs. Dès lors, il est possible d'élaborer une présentation standard pour aider les utilisateurs dans leur navigation. Il faut cependant que cette standardisation n'aille pas à l'encontre de la liberté de composition des pages WEB. Ainsi, si l'on décide d'utiliser des icônes, une bonne solution pourrait être de décider du type d'icône et non pas de fournir l'icône. Par exemple, pour représenter la catégorie "informatique", on pourrait demander d'utiliser une icône représentant un ordinateur et non pas d'utiliser telle icône représentant tel ordinateur.

En ce qui concerne l'apport d'information sur les liens, les moyens mis en œuvre sont peu présents. L'initiative de la Bacob d'utiliser une fenêtre de navigation est intéressante et devrait se généraliser.

6. PRESENTATION ACTUELLE DES LIENS

Afin de distinguer les liens du texte normal et d'attirer l'attention de l'utilisateur, les navigateurs les présentent dans un style différent (ex : par défaut, Netscape affiche les liens en bleu souligné tant qu'ils n'ont pas été visités et en magenta souligné lorsqu'ils l'ont été) et transforment le curseur de la souris en une petite main lorsque la souris passe sur un lien.

Actuellement, les moyens permettant de fournir de l'information sur les documents référencés par les liens sont limités. Généralement, le peu de texte placé près du lien ou le nom du lien ne suffisent pas pour donner une idée précise du contenu du document.

Il est possible d'utiliser les infobulles qui s'affichent lorsque le curseur reste un instant sur le lien. Celles-ci peuvent contenir du texte, mais bien souvent, parce que les créateurs des pages l'ont décidé ainsi, elles ne contiennent qu'un mot ou reprennent simplement l'adresse pointée par le lien. Une autre façon d'ajouter de l'information est d'utiliser Javascript, un langage de script qui permet, entre autres, de faire défiler du texte dans la barre sous l'application.

Enfin, une autre façon d'apporter de l'information est d'avoir recours aux frames. Elles permettent de découper l'écran du navigateur en plusieurs documents HTML. On peut ainsi afficher, dans une frame, des informations concernant le lien sur lequel se trouve le curseur de la souris. Les frames sont très commodes car les informations que l'on y place ne sont pas limitées en taille : si le texte est trop grand pour être affiché en entier, on peut le faire défiler.

7. PROPOSITIONS POUR UNE PRESENTATION

Les améliorations suivantes concernent la présentation des pages WEB, c'est-à-dire la façon dont les informations seront organisées et montrées à l'utilisateur. Ces améliorations peuvent être apportées soit séparément, soit en groupe, en veillant à ne pas surcharger l'utilisateur. Certaines améliorations nécessitent une intervention de la part du créateur de la page source, d'autres nécessitent une connexion à la page référencée. Si ces deux problèmes pouvaient être résolus, automatisés, les améliorations permettraient une meilleure vue d'ensemble des liens.

7.1.1. Utilisation d'Icônes

Devant chaque lien, on pourrait placer une icône représentative de la catégorie de document référence (fichiers son, vidéo, html...), du sujet général (ex : informatique, littéraire, éducatif, loisir, ...), de la localisation de la page (référence sur la page, serveur local, serveur externe) ou de la date de la dernière mise à jour (les icônes "NEW" sont souvent utilisées). Cette méthode ne pourrait pas être utilisée lorsque le lien fait partie d'un texte car elle gênerait la lisibilité de celui-ci. Toutefois, une icône est parfaitement adaptée lorsqu'il s'agit d'une énumération de liens placées les uns en dessous des autres.

Un autre problème vient du fait que c'est généralement au créateur de la page contenant le lien à ajouter les icônes. Si l'on pouvait automatiser le placement de l'icône (par exemple au moyen de champs dans le code HTML), cela permettrait une standardisation de la présentation des pages et fournirait ainsi une aide précieuse à la navigation.

La Figure V.10 donne un exemple d'icônes pouvant être employées pour indiquer le sujet ou la catégorie du document. Ces icônes proviennent des librairies standards de Windows et sont donc bien connues des utilisateurs du système d'exploitation de Microsoft.



Figure V.10 : Exemples d'icônes

7.1.2. Style de caractères

Pour pallier à l'inconvénient de l'icône, à savoir la réduction de la lisibilité, on pourrait changer le style de caractères (changement de police de mise en forme ou de couleur) pour indiquer la catégorie, la proximité et l'ancienneté. Cependant, il faut veiller à ne pas entrer en conflit avec les couleurs choisies par le navigateur pour indiquer les liens et à ne pas surcharger le document de couleurs ce qui aurait pour effet de désorienter l'utilisateur. A ce propos, un maximum de trois couleurs (sans les couleurs du navigateur) et de trois polices de caractères bien distinctes nous paraît un bon nombre (qui, de plus, répond à la règle ergonomique¹ du 5 ± 2). Voici un exemple de styles de caractères :

- LIEN POUR LA CATÉGORIE INFORMATIQUE :
- *lien pour la catégorie littéraire ;*
- **lien pour la catégorie loisir ;**

¹ Tiré de [Vanderdonkt]

Toutefois, il ne faut pas axer la différenciation des caractères seulement sur les couleurs car elles ne sont d'aucune aide pour les personnes ayant des problèmes oculaires (les daltoniens, par exemple).

7.1.3. Infobulles

La syntaxe actuelle du langage HTML permet l'utilisation d'infobulles, des bulles qui apparaissent sous le lien dès que l'on laisse le curseur de la souris sur le lien. Mais, jusqu'ici, on n'y trouve souvent qu'un mot ou l'adresse pointée par le lien, ce qui n'est pas d'une grande aide pour l'utilisateur. Au lieu de cela, on pourrait accoler une bulle d'aide contenant les mots clés de la page de destination, ceux-ci seraient indiqués par le créateur de la page. On pourrait prévoir un ajout à la syntaxe du langage HTML qui permettrait d'inclure une zone de mots clés ; il serait ainsi possible d'aller automatiquement les chercher pour les inscrire dans les infobulles.

Par exemple, l'infobulle d'un lien pointant sur une page traitant de JAVA pourrait contenir les mots suivants : caractéristiques, didacticiel, exemples...

7.1.4. Barre sous l'application

Dans le même ordre d'idée, on pourrait utiliser la barre sous l'application : en passant sur un lien, on verrait apparaître dans cette barre quelques mots clés ou une phrase expliquant le contenu de la page référencée. Encore une fois, ceci serait possible moyennant un ajout à la syntaxe du langage HTML ou en utilisant Javascript.

Toutefois, il faut veiller à ne pas utiliser toute la barre sous l'application car c'est à cet endroit qu'est indiquée l'adresse exacte lorsque le curseur de souris passe sur un lien. Cette information est importante et doit rester présente car, comme nous l'avons vu à la section 4, il est possible d'en tirer des informations.

7.1.5. Menu contextuel

Les menus contextuels (que l'on obtient généralement au moyen du bouton droit de la souris) pourraient contenir un item supplémentaire indiquant les propriétés du lien (date de création, de mise à jour, taille du document...). Ceci éviterait des visites sur des pages déjà connues et non modifiées et permettrait également de gagner du temps (par exemple, si l'on dispose de peu de bande passante, on ne visitera pas une page contenant beaucoup d'images).

7.1.6. Aperçu du document

Une dernière méthode pourrait être de faire appel à une fenêtre comportant un aperçu du document. Cet aperçu pourrait consister en une image de la page ou en la visualisation de l'ensemble des titres de la page. Il serait bien de pouvoir paramétrer une telle fonction afin de répondre aux besoins de l'utilisateur (n'afficher que les titres, que les images...). Bien entendu, cet aperçu nécessite que l'on charge la page, ce qui n'est pas le cas avec les autres techniques proposées.

8. CONCLUSION

Au cours de sa navigation, l'utilisateur du World Wide Web est souvent surchargé et désorienté. Face à un lien nouveau, il ne peut retirer que peu d'informations quant au contenu du document qu'il référence. Seule une analyse de l'adresse, ou les techniques actuelles de frames, infobulles et barre sous l'application peuvent l'aider. Une représentation graphique soit au moyen d'objets en deux dimensions (comme les graphes), soit au moyen de métaphores (représentées par les objets en trois dimensions) aide à mieux visualiser la structure du WEB mais elle est souvent dépassée par la taille des sites et la complexité des relations entre les pages. Un outil d'aide à la navigation et d'analyse sémantique faciliterait les recherches et diminuerait le parcours. De plus, il est possible, soit en modifiant la syntaxe du langage HTML, soit en chargeant au préalable les pages, d'apporter quelques améliorations à la présentation des documents qui, en fournissant plus d'informations sur les liens, auront pour effet de réduire la désorientation de l'utilisateur.

Conclusion

Quotidiennement, de nouveaux sites sont ajoutés sur le World Wide Web. Ces sites contiennent de plus en plus d'informations regroupées sur différentes pages. Pour réaliser ces pages, les créateurs disposaient, jusqu'il y a peu, uniquement du langage HTML, qui permet de présenter et de structurer les informations (hypertexte, images, fichiers...) grâce à une syntaxe simple basée sur un système de marques. Cependant, HTML ne permettait pas d'inclure et de jouer des sons et des vidéos, ni de lancer des applications.

Ce manque fut comblé, dès 1995, avec l'arrivée de JAVA. Ce langage de programmation orienté-objet, permet l'amélioration de pages WEB en y insérant des programmes appelés applets. Mais l'utilisation de Java ne se limite pas à cela : ses mécanismes de sécurité et sa portabilité en font le langage idéal pour réaliser des applications fonctionnant sur les réseaux. Nous avons présenté ces deux langages dans la première partie de ce mémoire.

En JAVA, nous avons développé une application, ChercheWeb, de parcours des pages Web et de reconnaissance des liens. Celle-ci est toute simple mais, si on lui apporte quelques modifications, elle permet d'identifier et de présenter graphiquement les liens contenus sur une page WEB. Nous avons présenté l'application et ses améliorations dans la deuxième partie.

Dans la troisième partie, nous nous sommes intéressés aux liens en tant que tels. Nous en avons tout d'abord tracé une typologie et nous avons considéré quelques moyens de représentation. Ensuite, nous avons vu comment retirer de l'information à partir d'une adresse. Puis, sur base de l'examen de quelques pages WEB, nous avons considéré les moyens actuels de présentation de liens. Pour terminer, nous avons donné quelques pistes concernant cette présentation.

Glossaire

ACTIVEX

ActiveX est l'architecture développée par la société Microsoft qui permet à des composants logiciels d'interagir sur les réseaux en utilisant n'importe quel langage.

API

API est l'abréviation de Application Programmable Interface. C'est une interface pour le développement d'applications.

APPLET

Les applets sont des programmes écrits en JAVA qui nécessitent un contexte pour s'exécuter, par exemple l'AppletViewer ou un navigateur. Les applets sont généralement insérées dans des pages WEB.

APPLETVIEWER

L'appletViewer est un programme créé par Sun fournissant un contexte permettant d'exécuter les applets sans l'aide d'un navigateur.

APPLICATION

Les applications sont des programmes écrits en JAVA qui peuvent être exécutées directement grâce à l'interpréteur JAVA sans nécessiter de navigateur.

AUTO GARBAGE COLLECTION

L'auto Garbage Collection est un mécanisme libérant périodiquement la mémoire des objets qui ne sont plus utilisés ou référencés.

AWT

L'AWT (Abstract Windowing Toolkit) est l'ensemble des classes permettant de créer des interfaces graphiques.

BOOKMARK

Voir signet.

BROWSER

Voir navigateur.

BYTE-CODE

Code JAVA compilé, indépendant de la plate-forme et qui sera exécuté par un interpréteur.

C++

Le C++ est un langage de programmation orienté objet.

CERN

Le CERN, Centre Européen de Recherche Nucléaire, est un des centres à l'origine du développement du World Wide Web.

CGI

Un CGI, Common Gateway Interface, est l'Interface définissant la communication entre une entité externe et un serveur WEB.

CLASSE

Une classe est un ensemble de variables et de méthodes qu'un objet peut utiliser.

COOKIE

Un cookie est envoyé par un serveur vers un navigateur et reprenant des informations sur la connexion à ce serveur, il est mis à jour à chaque nouvelle connexion à ce serveur.

CORBA

Common Object Broker Request Architecture, Corba est le standard industriel de développement de logiciels hétérogènes.

ENCAPSULATION

Technique des langages orientés objets qui consiste à déclarer les variables et les méthodes d'une classe de manière publique pour qu'elles puissent être utilisées par les autres classes sans connaître le code qui est privé.

ENVIRONNEMENT D'EXÉCUTION

L'environnement d'exécution de JAVA, en anglais JAVA Runtime Environment, contient les outils nécessaires à l'exécution de programmes JAVA.

ESPACE DE NOMS

Un espace de noms définit les portions de la Machine Virtuelle JAVA auxquelles peut accéder une applet.

FIREWALL

Un firewall est un ensemble de moyens matériels et logiciels mis en place pour garantir la sécurité d'un site contre des attaques provenant de l'extérieur.

FISH-EYE VIEW

Le fish-eye view est une technique de représentation des hiérarchies de liens en deux dimensions.

FRAME

Une frame est une fenêtre contenant un document WEB. On utilise les frames pour découper l'écran du navigateur en plusieurs documents HTML

FTP

FTP, File Transfer Protocole, est un protocole qui permet de transférer des fichiers via les réseaux.

GOPHER

Gopher est un protocole qui permet d'effectuer des recherches et de la classification sur les réseaux.

HÉRITAGE

Propriété des langages orientés objets qui permet à une classe d'utiliser les méthodes et variables de classes plus générales dont elle est une spécialisation.

HOTJAVA

Navigateur WEB écrit en langage JAVA par la société JAVASOFT

HTML

HTML, HyperText Markup Language, est le langage utilisé pour créer des pages WEB et est une implémentation de SGML.

HTTP

HTTP, HyperText Transfert Protocol, est le protocole utilisé pour transférer de l'information HTML.

HYPERBOLIC BROWSER

L'hyperbolic Browser est une adaptation de la technique de représentation des liens fish-eye view.

HYPERMEDIA

Hypermédia est la combinaison des termes Hypertexte et Multimédia.

HYPERTEXTE

Document de type texte contenant un lien vers un autre document.

IETF

L'IETF, Internet Engineering Task Force, est l'organisme qui a standardisé le langage HTML.

INFOBULLE

Une infobulle est une bulle d'aide qui apparaît dès que l'on laisse le curseur de la souris sur un lien.

INTERFACE

Ensemble de variables et de méthodes qui peuvent être implémentées par d'autres classes.

INTERPRÉTEUR

Programme effectuant la traduction d'un langage et permettant d'en exécuter le code.

IRC

IRC, Internet Relay Chat, est un service de communication textuelle multi-usagers.

JAR

JAR est l'abréviation de JAVA Archive, c'est un fichier qui contient le code de l'application JAVA ainsi que les fichiers qui s'y rapportent.

JAVA

Langage orienté objet créé par la société Sun permettant de développer des applets et des applications.

JAVA IDL

JAVA IDL, fournit une interopérabilité avec CORBA.

JAVABEANS™

JavaBeans est une API indépendante de la plate-forme et de l'architecture qui permet de créer et d'utiliser des composants dynamiques JAVA

JAVAS

JavaOS est le système d'exploitation optimisé pour JAVA et développé par la société JavaSoft.

JAVASCRIPT

JavaScript est une API qui permet la programmation d'événements, d'objets et d'actions au moyen d'un langage par script.

JAVASOFT

JavaSoft est une des sociétés de Sun s'occupant du développement des applications JAVA.

JDBC™

JDBC, JAVA Data Base Connectivity, est une interface SQL standard permettant l'accès et l'utilisation de bases de données aux programmes JAVA.

JDK

Voir Kit de Développement.

JFC

JFC, JAVA Foundation Classes, est un ensemble de classes permettant de créer des interfaces graphiques. Il est basé sur l'AWT qu'il remplace.

JNDI™

JNDI, JAVA Naming and Directory Interface, fournit un standard permettant des accès unifiés aux différents services de "naming and directory" au sein d'une entreprise.

JRE

Voir Environnement d'exécution.

KIT DE DÉVELOPPEMENT

Le kit de développement JAVA, en anglais JAVA Development Kit, est fourni par Sun et contient tous les outils nécessaires à la création et à l'exécution de programmes JAVA.

LIBRAIRIE DE CLASSES

Une librairie de classes, en anglais package, est un ensemble de librairies regroupées selon un thème commun.

LIVECONNECT

LiveConnect est une architecture fournie par Netscape qui permet la communication entre des applets JAVA et Javascript.

MACHINE VIRTUELLE JAVA

Une machine virtuelle JAVA, en anglais JAVA Virtual Machine, est un logiciel, implémenté au dessus du système d'exploitation de l'ordinateur sur lequel le programme sera exécuté, qui va simuler sur cet ordinateur le fonctionnement de la machine JAVA, rendant possible l'exécution du programme.

MAILTO

Mailto permet d'accéder au courrier électronique.

MÉTHODE NATIVE

Une méthode native est une méthode implémentée dans un autre langage que JAVA, généralement pour dialoguer directement avec le système d'exploitation et pour ainsi améliorer les performances.

MICROSOFT EXPLORER

Microsoft Explorer est le navigateur WEB développé par Microsoft.

NAVIGATEUR

Un navigateur, en anglais browser, est un logiciel utilisé pour parcourir les pages WEB. Les plus utilisés sont Netscape et Microsoft Explorer.

NAVIGATIONAL VIEW BUILDER

Représentation en deux dimensions de liens hypermédias.

NETSCAPE

Netscape est le nom de la société ayant développé un des navigateurs WEB les plus utilisés et par extension, ce navigateur.

NEWS

News permet d'accéder aux groupes de discussions sur Internet.

OAK

Oak est le nom de code du langage à l'origine de JAVA.

OBJET

Un objet est l'instanciation d'une classe.

OLE

OLE, Object Linking and Embedding, est une technique développée par Microsoft qui permet à une application de créer des documents contenant des composants d'autres applications.

OPENDOC

OpenDoc est une architecture d'API pour le développement d'applications multiplateformes et distribuées.

PACKAGE

Voir librairie de classes.

PAGE WEB

Une page WEB est un document réalisé en HTML, utilisé sur le WEB pour présenter des informations (hypertexte, sons, images, vidéo...).

PLUG-IN

Un plug-in est un petit logiciel qui ajoute de nouvelles possibilités aux navigateurs. Un plug-in realaudio, par exemple, permet au navigateur de jouer les fichiers realaudio.

POLYMORPHISME

Propriété des langages orientés objets qui permet à une méthode d'être déclarée avec des arguments de types différents.

PROGRAMME JAVA

Un programme JAVA est soit une applet, soit une application.

REALAUDIO

Realaudio est un type de fichier audio en temps réel.

SANDBOX

Une sandbox est une zone de mémoire allouée à une application, elle comprend un certain nombre de composants des systèmes coopératifs, allant du gestionnaire de sécurité, qui s'exécute comme une partie du programme, aux mesures de sécurité comprises dans la Machine Virtuelle JAVA et dans le langage lui-même

SERVEUR

Un serveur est un ordinateur qui fournit des services particuliers aux ordinateurs s'y connectant.

SGML

(Standard Generalize Markup Language) Document dont la structure est définie par des délimiteurs.

SITE WEB

Un site WEB est une ensemble de pages HTML et de fichiers se trouvant sur un serveur.

SIGNET

Un signet, en anglais bookmark, permet de marquer des pages intéressantes dans un navigateur.

TCP/IP

TCP/IP, Transmission Control Protocol/Internet Protocol, est un ensemble de protocoles utilisé par les réseaux (dont Internet).

TELNET

Telnet est un service d'émulation permettant de se connecter à distance sur un serveur.

THREAD

Un thread est un processus qui peut s'exécuter en parallèle avec d'autres processus.

URL

Un URL, Uniform Resource Locators, représente l'adresse exacte d'une ressource sur Internet.

W3 (PROJET W3)

Projet de développement du World Wide Web.

WEB

voir World Wide Web.

WORLD WIDE WEB

Le World Wide Web, est l'ensemble des ressources qui peuvent être accessibles au moyen de Gopher, FTP, Telnet, HTTP...

Références

REFERENCES BIBLIOGRAPHIQUES

- [Bray] T. Bray
Measuring the WEB
Proceedings of the Fifth International WWW Conference, 1996.
- [Byte 97a] **JAVA Security and Type Safety**
Byte, Janvier 1997, page 63
- [Byte 97b] **Today the WEB, Tomorrow the World**
Byte, Janvier 1997, page 68
- [Campione] Mary Campione, Kathy Walrath
The JAVA Tutorial, Object-Oriented programming for the Internet
Addison Wesley, 1996
- [Card] S.K. Card, G.R. Robertson, W. York
The WEBBook and the WEBForager : An Information Workspace for the World-Wide WEB
Proceedings of CHI '96, Avril 1996
- [Lamping] J. Lamping, R. Rao, P. Pirolli
A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies
Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, ACM, Mai 1995
- [Mukherjea] S. Mukherjea, J.D.Foley
Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views
Proceedings of ACM SIGCHI '95, Mai 1995

- [Newman] Alexander Newman et al.
Le Macmillan de JAVA
Simon & Shuster Macmillan, Paris 1996.
- [Rouard] M. Rouard, L. Goffinet
HyperNavi : Etat de l'art
Institut d'Informatique, FUNDP, Namur, 1996
- [Serpe] V. Serpe
Projet HyperNavi Phase 1 - Définition d'une typologie des liens hypermédias
Institut d'Informatique, FUNDP, Namur, 1996
- [Vanderdonckt] J. Vanderdonckt
Guide ergonomique des interfaces homme-machine
Presse Universitaires de Namur, 1994
- [Yuwono] Budi Yuwono, Savio L. Y. Lam, Jerry H. Ying, et Dik L. Lee.
A World Wide Web Resource Discovery System.
Proceedings of the Forth International Conference, 1995.

REFERENCES ELECTRONIQUES

- [Abelcom] Site WEB Abelcom
<http://www.abelcom.net/abelcom/fr/index.htm>
- [Arcadis] Site WEB Arcadis
<http://www.arcadis.be>
- [Bacob] Site WEB bacob
<http://www.bacob.be>
- [BBL] Site WEB BBL
<http://www.bbl.be>
- [Berners-Lee] T. Berners-Lee, D. Connolly.
Hypertext Markup Language - 2.0 Internet Draft.
<http://www.w3.org/hypertext/WWW/Markup/MarkUp.html>

- [CGER] Site WEB CGER
<http://www.cger.be>
- [Crédit] Site WEB Crédit Communal
<http://www.creditcommunal.be>
- [CWRU] Case Western Reserve University
Introduction to HTML
<http://www.cwru.edu/help/introHTML/Toc.html>
- [Cyberimmo] Site WEB Cyberimmo
<http://www.cyberimmo.com>
- [Fritzinger] J. Steven Fritzinger, Marianne Mueller
JAVA Security
Sun Microsystems Computer Company, 1995 (<http://java.sun.com>)
- [Gate71] Site WEB Gate 71
<http://www.gate71.be>
- [Gbank] Site WEB Générale de Banque
<http://www.gbank.be>
- [Graham] Ian Graham
Introduction to HTML and URL's, University of Toronto
<http://www.utoronto.ca/WEBdocs/HTMLdocs/NewHTML/intro.html>
- [Gosling] James Gosling, Henry McGilton
The JAVA Language Environment, A Whitepaper
Sun Microsystems Computer Company, 1995 (<http://java.sun.com>)
- [Immobel] Site WEB Immobel
<http://www.immobel.com>
- [Immobilier] Site du WEB de l'immobilier
<http://immobilier.ami.fr>

- [Interpac] Site WEB Interpac
<http://www.interpac.be>
- [Maisons] Site WEB Maisons de la Côte
<http://www.easynet.fr/maisons/>
- [NCSA] NCSA National Center for Supercomputing Applications
The ABC's of HTML
<http://www.ncsa.uiuc.edu/General/Training/HTMLintro/Intro.html>
- [Ping] Site WEB Ping
<http://www.ping.be>
- [Sandia] Sandia National Laboratories
HTML Reference Manual
http://www.sandia.gov/sci-compute/html_ref.html
- [Skynet] Site WEB Skynet
<http://www.skynet.be>
- [Sun 1] Sun Microsystems, Inc.
Secure computing with JAVA™ : Now and the Future
<http://www.javasoft.com/marketing/collateral/security.html>
- [Sun 2] Sun Microsystems, Inc.
JAVA™ Foundation Classes : now and the Future
<http://www.javasoft.com/marketing/collateral/foundation-classes.html>
- [Sun 3] Sun Microsystems, Inc.
JAVA™ Foundation Classes
<http://www.javasoft.com/products/jfc/index.html>
- [Sun 4] Sun Microsystems, Inc.
JAVA API Overview & Schedule
<http://www.javasoft.com/products/API-overview/index.html>

- [Sun 5] Sun Microsystems, Inc.
Products & APIs
<http://www.javasoft.com/products/index.html>
- [Sun95] Sun Microsystems, Inc.
The JAVA™ Language : An Overview
Sun Microsystems Computer Company, 1995 (<http://java.sun.com>)

Annexes

Annexe 1:
Implémentation de l'Application ChercheWeb

Afin de réaliser l'implémentation de l'application ChercheWeb, il fut nécessaire de créer les six classes suivantes :

- | | |
|--------------------|--------------------|
| 1. Classe ChercheW | 2. Classe Afficher |
| 3. Classe Quitter | 4. Classe Apropos |
| 5. Classe Cherche | 6. Classe Histo |

Nous allons à présent les expliquer une à une en nous attardant sur la classe Cherche qui est le coeur de l'application, les autres classes étant des classes d'interface.

CLASSE CHERCHEW

Déclaration : `public class ChercheW extends Frame`

But : Créer le menu et les sous-menus de l'application ChercheWeb.

Méthodes utilisées :

- `public ChercheW()` : méthode d'initialisation de l'interface de l'application ChercheWeb. Cette méthode crée tous les composants de l'interface et les place.
- `public boolean handleEvent(Event event)` : méthode de réaction à l'événement de fermeture de l'application.
- `public boolean action(Event event, Object arg)` : méthode de réaction aux actions sur les items de menu.
- `void miOuvrir_Clicked(Event event)` : méthode de réaction à l'événement item de menu "Ouvrir..." cliqué. Cette méthode permet d'afficher la fenêtre d'ouverture d'un fichier, permet à l'utilisateur de sélectionner le fichier à ouvrir et ensuite lance l'affichage du contenu du fichier.
- `void miRechercher_Clicked(Event event)` : méthode de réaction à l'événement item de menu "Rechercher..." cliqué. Cette méthode permet de créer et d'afficher une fenêtre "Rechercher", instance de la classe "Cherche".
- `void miSave_Clicked(Event event)` : méthode de réaction à l'événement item de menu "Enregistrer..." cliqué. Cette méthode permet à l'utilisateur d'enregistrer, dans un

fichier texte, le contenu de la dernière recherche effectuée. Les informations contenues dans le fichier sont l'adresse de la page visitée, la date de la recherche et le contenu de la page visitée.

- void miQuitter_Clicked(Event event) : méthode de réaction à l'événement item de menu "Quitter..." cliqué. Cette méthode permet de créer et d'afficher une fenêtre "Quitter", instance de la classe "Quitter".
- void mihisto_Clicked(Event event) : méthode de réaction à l'événement item de menu "Historique..." cliqué. Cette méthode permet de créer et d'afficher une fenêtre "Historique", instance de la classe "Histo".
- void miapropos_Clicked(Event event) : méthode de réaction à l'événement item de menu "A propos de ..." cliqué. Cette méthode permet de créer et d'afficher une fenêtre "A propos", instance de la classe "Apropos".
- public static void main() : méthode principale. Cette méthode permet de créer et d'afficher une fenêtre "ChercheWeb" instance de la classe ChercheW.

Objets d'interface utilisés :

- Barre de Menu barre_menu
- Menus m_fichier, m_histo, m_about
- Items de Menu mi_Ouvrir, mi_Rechercher, mi_Save, mi_Quitter, mi_histo, mi_apropos

La Figure A.1 représente le menu principal de l'application :

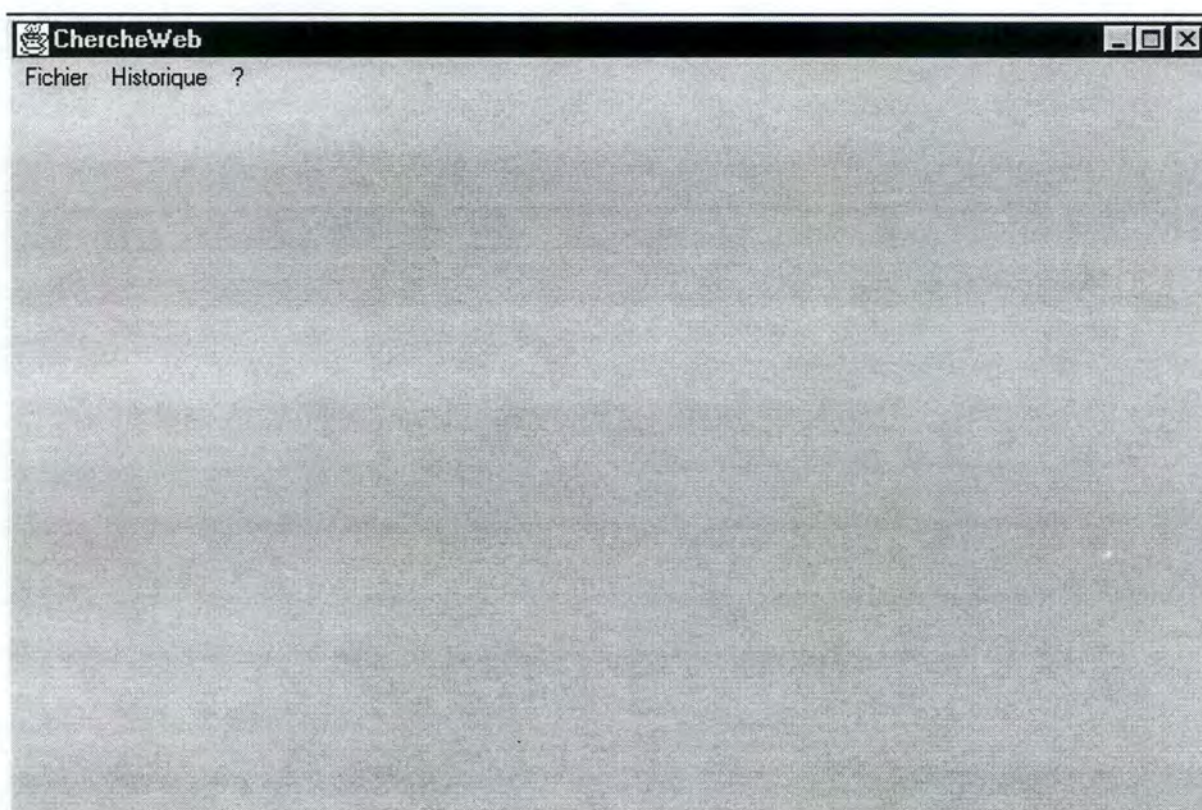


Figure A.1 : Le menu de l'application ChercheWeb

CLASSE AFFICHER

Déclaration : `class Afficher extends Dialog`

But : créer une fenêtre "Afficher" dans laquelle on affiche du texte.

Méthodes utilisées :

- `public Afficher(Frame parent, boolean modal)` : méthode d'initialisation de la fenêtre "Afficher". Cette méthode crée tous les composants de l'interface et les place.
- `public Afficher(Frame parent, String title, boolean modal)` : méthode permettant d'attribuer un titre à la fenêtre.
- `public synchronized void show(String message)` : méthode de placement et d'affichage de la fenêtre "Afficher".
- `public boolean handleEvent(Event event)` : méthode de réaction aux événements.
- `void btok_Clicked(Event event)` : méthode de réaction à l'événement bouton de commande "Ok" cliqué. Cette méthode permet à l'utilisateur de sortir de la fenêtre d'affichage.

Objets d'interface utilisés :

- Champ d'édition multilinéaire `txt_cont`
- Bouton de commande `bt_ok`

La Figure A.2 représente la fenêtre "Afficher" :

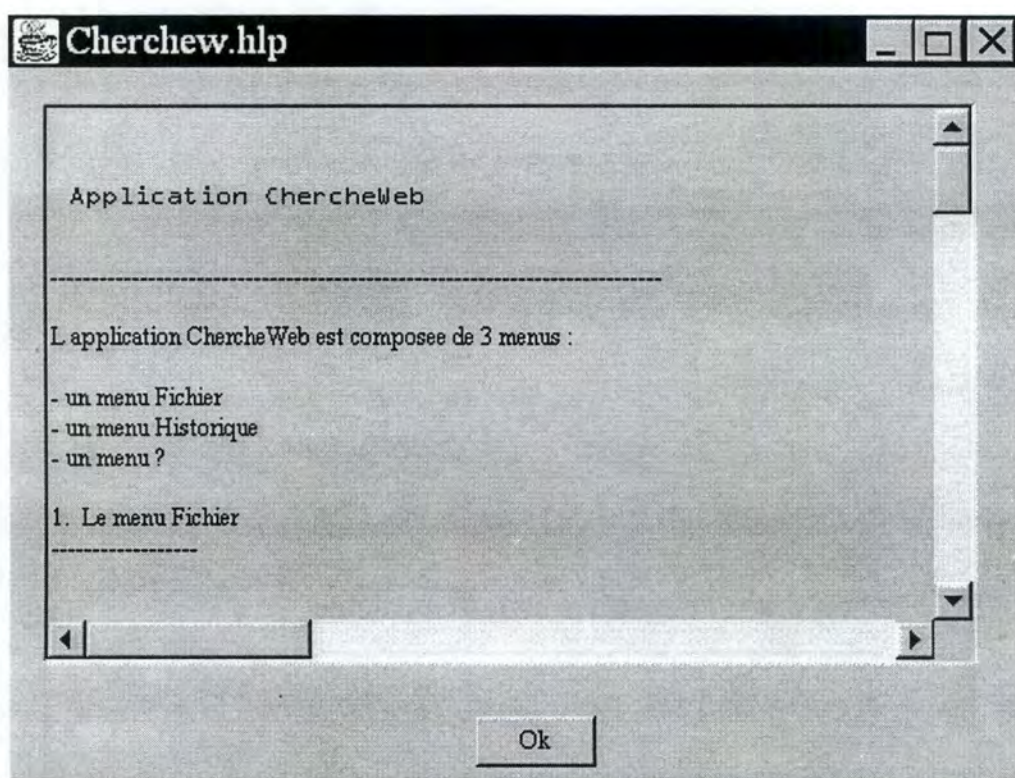


Figure A.2 : la fenêtre Afficher

CLASSE QUITTER

Déclaration : `class Quitter extends Dialog`

But : créer la fenêtre "Quitter" de sortie du programme.

Méthodes utilisées :

- `public Quitter(Frame parent, boolean modal)`: méthode d'initialisation de la fenêtre "Quitter". Cette méthode crée tous les composants de l'interface et les place.
- `public Quitter(Frame parent, String title, boolean modal)` : méthode permettant d'attribuer un titre à la fenêtre.
- `public synchronized void show(String message)` : méthode de placement et d'affichage de la fenêtre "Quitter".
- `public boolean handleEvent(Event event)` : méthode de réaction aux événements.
- `void btoui_Clicked(Event event)` : méthode de réaction à l'événement bouton de commande "Oui" cliqué. Cette méthode permet à l'utilisateur de sortir de la fenêtre d'affichage et de l'application.
- `void btnon_Clicked(Event event)` : méthode de réaction à l'événement bouton de commande "Non" cliqué. Cette méthode permet à l'utilisateur de sortir de la fenêtre d'affichage et de retourner à l'application.

Objets d'interface utilisés :

- Libellé label1
- Boutons de commande bt_oui, bt_non

La Figure A.3 représente la fenêtre "Quitter" :

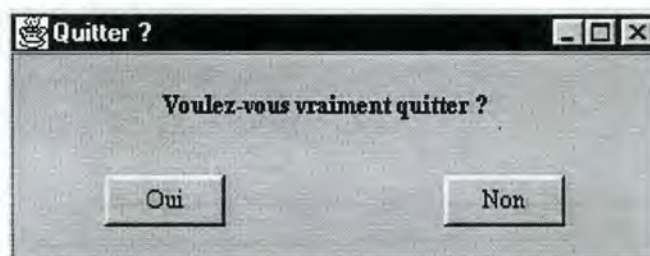


Figure A.3 : La fenêtre "Quitter"

CLASSE APROPOS

Déclaration : `class Apropos extends Dialog`

But : créer la fenêtre "A propos de ..." d'informations sur l'application ChercheWeb.

Méthodes utilisées :

- `public Apropos(Frame parent, boolean modal)`: méthode d'initialisation de la fenêtre "A propos". Cette méthode crée tous les composants de l'interface et les place.
- `public Apropos(Frame parent, String title, boolean modal)`: méthode permettant d'attribuer un titre à la fenêtre.
- `public synchronized void show(String message)`: méthode de placement et d'affichage de la fenêtre "A propos".
- `public boolean handleEvent(Event event)`: méthode de réaction aux événements.
- `void btOk_Clicked(Event event)`: méthode de réaction à l'événement bouton de commande Ok cliqué. Cette méthode permet à l'utilisateur de sortir de la fenêtre "A propos".

Objets d'interface utilisés :

- Libellés label1, label2, label3, label4, label5, label6
- Bouton de commande bt_Ok

La Figure A.4 représente la fenêtre "A propos de..." :

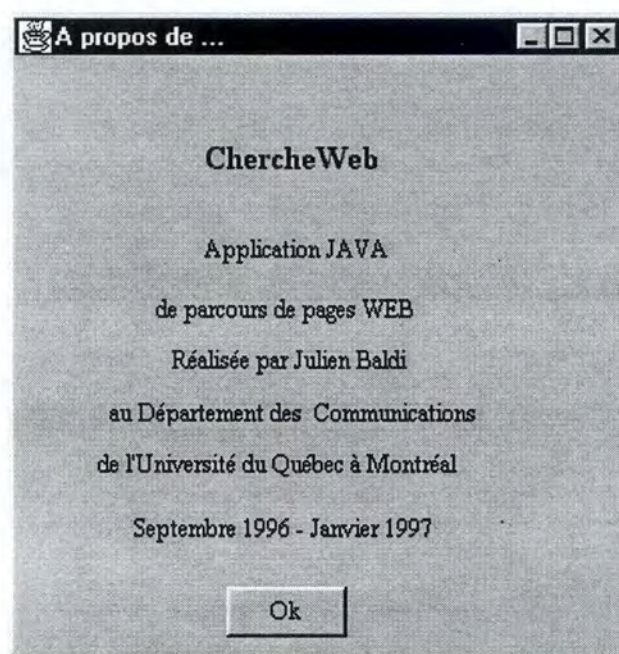


Figure A.4 : La fenêtre "A propos de ..."

CLASSE CHERCHE

Déclaration : `class Cherche extends Dialog implements Runnable`

But : créer la fenêtre "Cherche" de recherche sur le Web.

Méthodes utilisées :

- `public Cherche(Frame parent, boolean modal)` : méthode d'initialisation de la fenêtre "Cherche". Cette méthode crée tous les composants de l'interface et les place.
- `public Cherche (Frame parent, String title, boolean modal)` : méthode permettant d'attribuer un titre à la fenêtre.
- `public synchronized void show(String message)` : méthode de placement et d'affichage de la fenêtre "Cherche".
- `public boolean handleEvent(Event event)` : méthode de réaction aux événements.
- `public void start ()` : méthode permettant d'initialiser le multithreading.
- `public void stop ()` : méthode permettant de stopper le multithreading.
- `public void run ()` :
- `void btquit_Clicked(Event event)` :
- `void btToc_Clicked(Event event)` :
- `void btRechercher_Clicked(Event event)` : méthode de réaction à l'événement bouton de commande "Rechercher" cliqué. Cette méthode permet de lancer la recherche sur l'adresse indiquée dans le champ d'édition.
- `void btStop_Clicked(Event event)` : méthode de réaction à l'événement bouton de commande "Stop" cliqué. Cette méthode permet de stopper la recherche.
- `void txtUrlIn_EnterHit(Event event)` : méthode permettant de lancer la recherche sur l'adresse indiquée dans le champ d'édition lorsqu'on appuie sur la touche [Enter].

- `void btEffacer_Clicked(Event event)` : méthode de réaction à l'événement bouton de commande "Effacer" cliqué. Cette méthode permet d'effacer le contenu du champ d'édition où l'on inscrit l'adresse de la page à consulter.
- `void btChercher_Clicked(Event event)` : méthode de réaction à l'événement bouton de commande "Chercher" cliqué. Cette méthode permettant de faire une recherche à partir d'un lien de la liste des liens de la page consultée.

Objets d'interface utilisés :

- Libellé `lbl_url_in`, `lbl_contenu`
- Champs d'édition unilinéaire `txt_url_in`, `txt_mess`
- Champ d'édition multilinéaire `txt_contenu`
- Boutons de commande `bt_rechercher`, `bt_effacer`, `bt_toc`, `bt_chercher`, `bt_stop`, `bt_quit`

Explication de la recherche

Avant toute chose, un thread est créé afin d'accélérer la recherche et de pouvoir la stopper quand on le désire. L'application se connecte alors à la page WEB dont l'utilisateur a donné l'adresse (si cette adresse est valide, sinon, un message d'erreur est renvoyé). L'application crée alors un nouveau flux de données (InputStream dans le langage JAVA), afin de pouvoir lire le contenu de la page WEB. Une fois cela fait, la recherche commence au sein de cette page WEB : tant qu'on n'est pas arrivé à la fin du fichier HTML, on vérifie ligne par ligne la présence de l'identificateur de lien `href="http://`. Si on trouve un tel identificateur, on copie ce qui vient après dans un tampon (ex : si on trouve `href="http://www.nom.suffixe/page.html"`, on place `www.nom.suffixe/page.html` dans le tampon). On considère que le lien est trouvé quand on atteint des guillemets. Lorsque l'on est à la fin du fichier, on ferme le flux de données, on copie le contenu du tampon dans la zone prévue pour l'affichage des liens.

La Figure A.5 représente la fenêtre "Cherche"

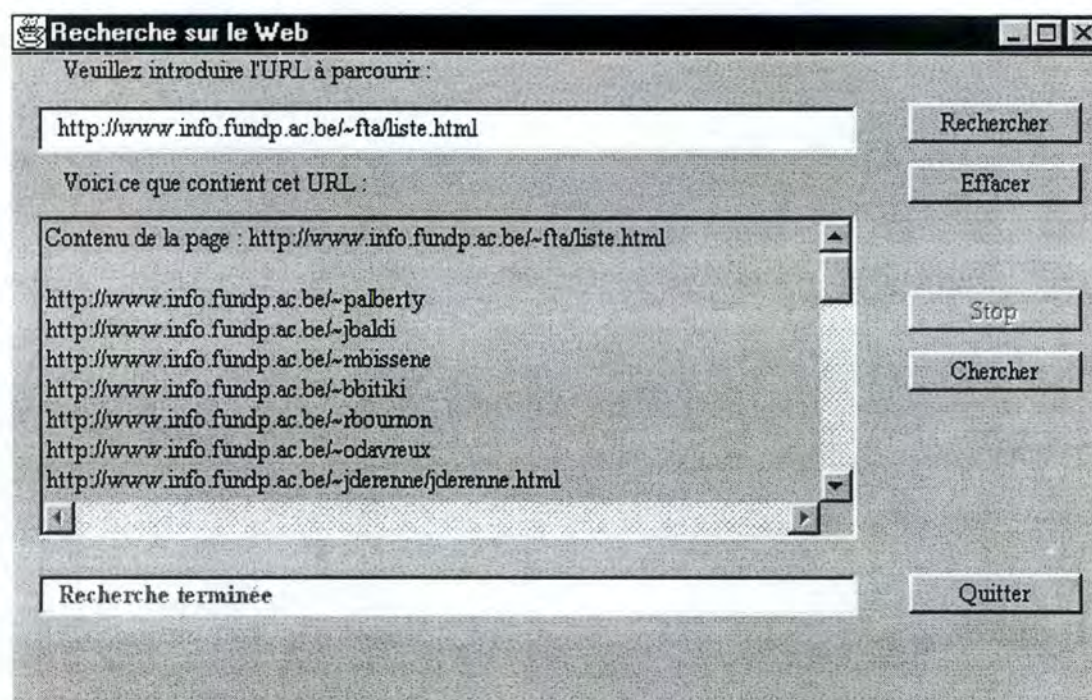


Figure A.5 : La fenêtre "Cherche"

CLASSE HISTO

Déclaration : `class Histo extends Dialog`

But : créer la fenêtre "Historique".

Méthodes utilisées :

- `public Histo(Frame parent, boolean modal)` : méthode d'initialisation de la fenêtre "Historique". Cette méthode crée et place les composants de l'interface.
- `public Histo(Frame parent, String title, boolean modal)` : méthode permettant d'attribuer un titre à la fenêtre.
- `public synchronized void show(String message)` : méthode de placement et d'affichage de la fenêtre "Historique".
- `public boolean handleEvent(Event event)` : méthode de réaction aux événements.
- `void btquitter_Clicked(Event event)` : méthode de réaction à l'événement bouton de commande Quitter cliqué, qui permet de sortir de la fenêtre.

Objets d'interface utilisés :

- Libellé `label1`
- Champ d'édition multilinéaire `txt_pages`
- Bouton de commande `bt_quitter`

La Figure A.6 représente la fenêtre "Histo" :

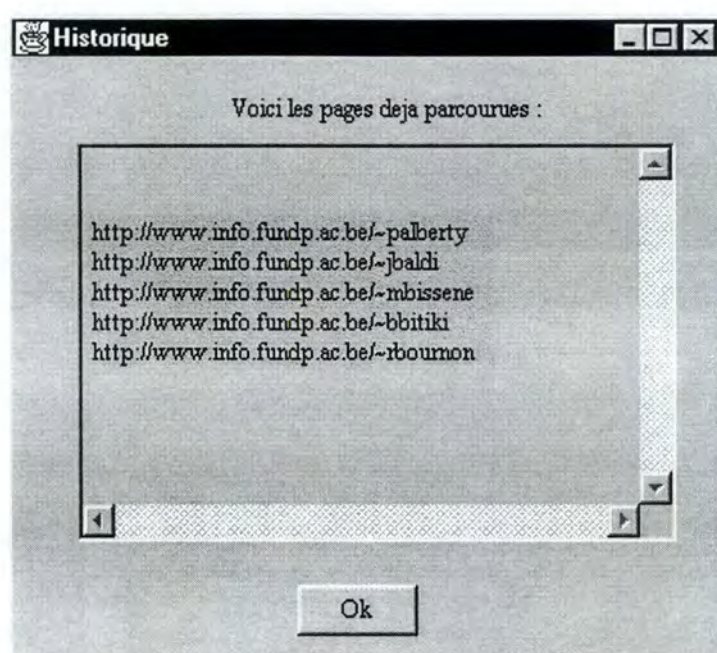


Figure A.6 : La fenêtre "Historique"

Annexe 2 :
Code de l'application ChercheWeb

/**

Application ChercheW - 27 janvier 1997

Cette application crée une barre de menus composée de 3 menus :
Fichier, Historique et ?.

Le but principal de cette application est de parcourir une page web et
d'en
sortir tous les liens qui se trouvent sur cette page.

REM : le fichier d aide est ChercheW.hlp

Réalisé par Julien Baldi - Janvier 1997.

Infos et commentaires : jbalidi@info.fundp.ac.be

Merci à Louis-Claude Paquin, Dany Beaupré et aux étudiants de Maîtrise en
Communication de L'Université du Québec à Montréal.

*/

```
import java.awt.*;
import java.net.*;
import java.io.*;
import java.util.Date;
```

```
public class Cherchew extends Frame
{
```

```
    boolean inAnApplet = true;
    private Quitter d_quitter;
    private Apropos d_about;
    private Cherche d_cherche;
    private Histo d_histo;
    private FileDialog d_save;
    private FileDialog d_ouvrir;
    private Afficher d_afficher;
```

```
    String LSEP = System.getProperty("line.separator");
    static String[] tab_ad = new String[20];
    static int lg_tabad = 0;
```

```
    public Cherchew()
    {
```

```
        setLayout(null);
        addNotify();
        reshape(15,30,600,400);
        setResizable(false);
```

```
        barre_menu = new java.awt.MenuBar();
        setMenuBar(barre_menu);
```

```
        m_fichier = new java.awt.Menu("Fichier", true);
        barre_menu.add(m_fichier);
        mi_Ouvrir = new java.awt.MenuItem("Ouvrir...");
        m_fichier.add(mi_Ouvrir);
        mi_Rechercher = new java.awt.MenuItem("Rechercher...");
```



```

m_fichier.add(mi_Rechercher);
m_fichier.addSeparator();
mi_Save = new java.awt.MenuItem("Enregistrer...");
mi_Save.disable();
m_fichier.add(mi_Save);
m_fichier.addSeparator();
mi_Quitter = new java.awt.MenuItem("Quitter...");
m_fichier.add(mi_Quitter);

m_histo = new java.awt.Menu("Historique", true);
mi_histo = new java.awt.MenuItem("Historique...");
m_histo.add(mi_histo);
barre_menu.add(m_histo);

m_about = new java.awt.Menu("?", true);
mi_apropos = new java.awt.MenuItem("A propos de ...");
m_about.add(mi_apropos);
barre_menu.add(m_about);
}

public boolean handleEvent(Event event)
{
    if (event.id == Event.WINDOW_DESTROY)
    {
        if (inAnApplet)
        {
            dispose();
        }
        else
        {
            System.exit(0);
        }
    }
    return super.handleEvent(event);
}

public boolean action(Event event, Object arg)
{
    if (event.target == mi_Ouvrir)
    {
        miOuvrir_Clicked(event);
    }
    if (event.target == mi_Save)
    {
        miSave_Clicked(event);
    }
    if (event.target == mi_Rechercher)
    {
        miRechercher_Clicked(event);
    }
    if (event.target == mi_Quitter)
    {
        miQuitter_Clicked(event);
    }
    if (event.target == mi_histo)
    {
        mihisto_Clicked(event);
    }
    if (event.target == mi_apropos)

```

```

    {
        miapropos_Clicked(event);
    }
    return true;
}

//{{{DECLARE_CONTROLS
java.awt.MenuBar barre_menu;
java.awt.Menu m_fichier;
java.awt.Menu m_histo;
java.awt.Menu m_about;
java.awt.MenuItem mi_Ouvrir;
java.awt.MenuItem mi_Rechercher;
static java.awt.MenuItem mi_Save;
java.awt.MenuItem mi_Quitter;
java.awt.MenuItem mi_histo;
java.awt.MenuItem mi_apropos;
//}}}}

void miOuvrir_Clicked(Event event)
{
    String fichier = "";
    String inputLine = "";

    String FSEP = System.getProperty("file.separator");
    String PSEP = System.getProperty("path.separator");
    String mess = "";

    d_ouvrir = new FileDialog(this, "Ouvrir...", 0);
    d_ouvrir.show();
    d_afficher = null;

    if (d_ouvrir.getFile() != null)
    {
        try
        {
            fichier = d_ouvrir.getDirectory() + FSEP +
                d_ouvrir.getFile();
            File inputfile = new File (fichier);
            FileInputStream fis = new FileInputStream (inputfile);
            DataInputStream dis = new DataInputStream(fis);
            while ((inputLine = dis.readLine()) != null)
            {
                mess = mess + LSEP + inputLine;
            }

            dis.close();
            fis.close();

            //System.out.println(mess);
            d_afficher = new Afficher(this, d_ouvrir.getFile(), true);
            d_afficher.show(mess);

        } // fin du try
        catch (FileNotFoundException e)
        {
            System.err.println("File not found: " + e);
        }
    }
}

```



```

    }
    catch (IOException e)
    {
        System.err.println("IO exception: " + e);
    }
}
/* else
{
    d_afficher = new Afficher(this, "Coucou", true);
    d_afficher.show("beuh");
}*/

}

void miRechercher_Clicked(Event event)
{
    d_cherche = new Cherche(this, "Recherche sur le Web", true);
    d_cherche.show();
}

void miSave_Clicked(Event event)
{
    if (d_cherche != null)
    {

        d_save = new FileDialog(this, "Enregistrer sous...", 1);
        d_save.show();

        String fichier = "";
        String inputLine = "";

        Date now = new Date();
        String d,t ;

        d = now.getDate() + "/" + (now.getMonth() + 1) + "/" + 19 +
            now.getYear() + " ";
        t = now.getHours() + ":" + now.getMinutes() + ":" +
            now.getSeconds();
        String temps = LSEP + LSEP + "Enregistre le " + d + "a "
            + t;

        String FSEP = System.getProperty("file.separator");
        String PSEP = System.getProperty("path.separator");
        String mess = "";

        fichier = d_save.getDirectory() + FSEP +
            d_save.getFile();

        if( d_save.getFile() != null)
        {
            mess = d_cherche.affichage;

            try
            {

                File outputfile = new File (fichier);
                FileOutputStream fos = new FileOutputStream
                    (outputfile);
            }
        }
    }
}

```

```

        DataOutputStream dos = new DataOutputStream(fos);

        dos.writeBytes(mess + temps);

        dos.close();
        fos.close();

    } // fin du try
    catch (FileNotFoundException e)
    {
        System.err.println("File not found: " + e);
    }
    catch (IOException e)
    {
        System.err.println("IO exception: " + e);
    }
    } // fin du if ( d_save.getFile() != null)

    } // fin du if (d_cherche != null)
}

void miQuitter_Clicked(Event event)
{
    d_quitter = new Quitter(this, "Quitter ?", true);
    d_quitter.show();
}

void mihisto_Clicked(Event event)
{
    if (d_histo == null)
    {
        d_histo = new Histo(this, "Historique", false);

        for (int i= 1; i <= lg_tabad; i++ )
        {
            d_histo.txt_pages.appendText(tab_ad[i] + LSEP);
        }

        d_histo.show();
    }
    else
    {
        d_histo.show();
    }
}

void miapropos_Clicked(Event event)
{
    d_about = new Apropos(this, "A propos de ...", true);
    d_about.show();
}

public static void main(String argv[])
{
    Cherchew window = new Cherchew();

```



```

        window.inAnApplet = false;
        window.setTitle("ChercheWeb");
        window.pack();
        window.show();
    }
}

// Creation de la boite de dialogue Afficher
class Afficher extends Dialog
{
    public Afficher(Frame parent, boolean modal)
    {
        super(parent, modal);

        //{{INIT_CONTROLS
        setLayout(null);
        addNotify();
        resize(insets().left + insets().right + 500, insets().top +
            insets().bottom + 350);
        txt_cont = new java.awt.TextArea(10,100);
        txt_cont.reshape(insets().left + 15, insets().top +
            15, 465, 280);
        txt_cont.setEditable(false);
        add(txt_cont);
        bt_ok = new java.awt.Button("Ok");
        bt_ok.reshape(insets().left + 230, insets().top + 320 , 60, 25);
        add(bt_ok);
        setResizable(false);
        //}}
    }

    public Afficher(Frame parent, String title, boolean modal)
    {
        this(parent, modal);
        setTitle(title);
    }

    //{{DECLARE_CONTROLS
    java.awt.TextArea txt_cont;
    java.awt.Button bt_ok;
    //}}

    public synchronized void show(String message)
    {
        Rectangle bounds = getParent().bounds();
        Rectangle abounds = bounds();

        move(bounds.x + (bounds.width - abounds.width)/ 2,
            bounds.y + (bounds.height - abounds.height)/2);

        txt_cont.setText(message);

        super.show();
    }

    public boolean handleEvent(Event event)

```

```

    {
        if(event.id == Event.WINDOW_DESTROY)
        {
            hide();
            return true;
        }
        if (event.target == bt_ok && event.id == Event.ACTION_EVENT) {
            btok_Clicked(event);
        }
        return super.handleEvent(event);
    }

    void btok_Clicked(Event event)
    {
        hide();
    }

}

// fin de la creation de la boite de dialogue Afficher

// Creation de la boite de dialogue Quitter
class Quitter extends Dialog
{
    public Quitter(Frame parent, boolean modal)
    {
        super(parent, modal);

        //{{INIT_CONTROLS
        setLayout(null);
        addNotify();
        resize(insets().left + insets().right + 320,insets().top +
            insets().bottom + 100);
        label1 = new java.awt.Label("Voulez-vous vraiment quitter ?");
        label1.reshape(insets().left + 40,insets().top + 11,240,25);
        label1.setFont(new Font("Dialog", Font.BOLD, 12));
        add(label1);
        bt_oui = new java.awt.Button("Oui");
        bt_oui.reshape(insets().left + 45,insets().top + 60,60,25);
        add(bt_oui);
        bt_non = new java.awt.Button("Non");
        bt_non.reshape(insets().left + 215,insets().top + 60,60,25);
        add(bt_non);
        setTitle("Quitter");
        setResizable(false);
        //}}
    }

    public Quitter(Frame parent, String title, boolean modal)
    {
        this(parent, modal);
        setTitle(title);
    }

    //{{DECLARE_CONTROLS
    java.awt.Label label1;
    java.awt.Button bt_oui;
    java.awt.Button bt_non;

```



```

    //}}

public synchronized void show()
{
    Rectangle bounds = getParent().bounds();
    Rectangle abounds = bounds();

    move(bounds.x + (bounds.width - abounds.width)/ 2,
        bounds.y + (bounds.height - abounds.height)/2);

    super.show();
}

public boolean handleEvent(Event event)
{
    if(event.id == Event.WINDOW_DESTROY)
    {
        hide();
        return true;
    }
    if (event.target == bt_oui && event.id == Event.ACTION_EVENT)
    {
        btoui_Clicked(event);
    }
    if (event.target == bt_non && event.id == Event.ACTION_EVENT)
    {
        btnon_Clicked(event);
    }
    return super.handleEvent(event);
}

void btoui_Clicked(Event event)
{
    System.exit(0);
}

void btnon_Clicked(Event event)
{
    hide();
}

} // fin de la creation de la boite de dialogue Quitter

//Creation de la boite de dialogue A propos de
class Apropos extends Dialog {

    public Apropos(Frame parent, boolean modal)
    {
        super(parent, modal);

        //{{INIT_CONTROLS
        setLayout(null);
        addNotify();
        resize(insets().left + insets().right + 300, insets().top +
            insets().bottom + 296);
        labell = new java.awt.Label("de parcours de pages WEB");
        labell.reshape(insets().left + 70, insets().top + 126, 176, 17);
        add(labell);
    }
}

```

```

        label2 = new java.awt.Label("Application JAVA");
        label2.reshape(insets().left + 94,insets().top + 94,122,22);
        add(label2);
        label0 = new java.awt.Label("ChercheWeb");
        label0.reshape(insets().left + 80,insets().top + 35,132,30);
        label0.setFont(new Font("Dialog", Font.BOLD, 16));
        add(label0);
        label3 = new java.awt.Label("Réalisée par Julien Baldi");
        label3.reshape(insets().left + 50,insets().top + 149,179,20);
        add(label3);
        label4 = new java.awt.Label("au Département des
            Communications");
        label4.reshape(insets().left + 47,insets().top + 178,246,14);
        add(label4);
        label5 = new java.awt.Label("Septembre 1996 - Janvier 1997");
        label5.reshape(insets().left + 59,insets().top + 231,205,23);
        add(label5);
        label6 = new java.awt.Label("de l'Université du Québec à
            Montréal");
        label6.reshape(insets().left + 41,insets().top + 200,254,22);
        add(label6);
        setTitle("A propos de...");
        bt_Ok = new java.awt.Button("Ok");
        bt_Ok.reshape(insets().left + 115,insets().top + 265,60,25);
        add(bt_Ok);
        setResizable(false);
        //}}
    }

    public Apropos(Frame parent, String title, boolean modal)
    {
        this(parent, modal);
        setTitle(title);
    }

    //{{DECLARE_CONTROLS
    java.awt.Label label0;
    java.awt.Label label1;
    java.awt.Label label2;
    //symantec.itools.awt.NervousText nervousText1;
    java.awt.Label label3;
    java.awt.Label label4;
    java.awt.Label label5;
    java.awt.Label label6;
    java.awt.Button bt_Ok;
    //}}

    public synchronized void show() {
        Rectangle bounds = getParent().bounds();
        Rectangle abounds = bounds();

        move(bounds.x + (bounds.width - abounds.width)/ 2,
            bounds.y + (bounds.height - abounds.height)/2);

        super.show();
    }

    public boolean handleEvent(Event event)
    {

```



```

        if(event.id == Event.WINDOW_DESTROY)
        {
            hide();
            return true;
        }
        if (event.target == bt_Ok && event.id == Event.ACTION_EVENT)
        {
            btOk_Clicked(event);
        }
        return super.handleEvent(event);
    }

    void btOk_Clicked(Event event)
    {
        hide();
    }
}

// fin de la creation de la boite de dialogue A propos de

// Creation de la boite de dialogue Cherche

class Cherche extends Dialog implements Runnable
{
    String affichage = "";
    String LSEP = System.getProperty("line.separator");

    //String[] tab_ad = new String[20];
    //int lg_tabad = 0;

    public Cherche(Frame parent, boolean modal)
    {
        super (parent, modal);

        //{{INIT_CONTROLS
        setLayout(null);
        addNotify();
        resize(550,350);
        lbl_url_in = new java.awt.Label("Veuillez introduire l'URL à
        parcourir :");
        lbl_url_in.reshape(17,21,402,19);
        add(lbl_url_in);
        txt_url_in = new java.awt.TextField();
        txt_url_in.reshape(16,48,409,25);
        add(txt_url_in);
        bt_rechercher = new java.awt.Button("Rechercher");
        bt_rechercher.reshape(450,47,86,20);
        add(bt_rechercher);
        lbl_contenu = new java.awt.Label("Voici ce que contient cet
        URL :");
        lbl_contenu.reshape(17,76,318,22);
        add(lbl_contenu);
        txt_contenu = new java.awt.TextArea(10,100);
        txt_contenu.setEditable(false);
        add(txt_contenu);
        txt_contenu.reshape(16,102,409,162);
        bt_effacer = new java.awt.Button("Effacer");
        bt_effacer.reshape(450,77,86,20);
        add(bt_effacer);
        bt_toc = new java.awt.Button("Titres");
    }
}

```

```

        bt_toc.reshape(450,200,86,20);
        //add(bt_toc);
        bt_chercher = new java.awt.Button("Chercher");
        bt_chercher.reshape(450,170,86,20);
        add(bt_chercher);
        bt_stop = new java.awt.Button("Stop");
        bt_stop.reshape(450,140,86,20);
        add(bt_stop);
        txt_mess = new java.awt.TextField();
        txt_mess.setEditable(false);
        txt_mess.disable();
        txt_mess.reshape(16,280,409,22);
        txt_mess.setFont(new Font("Dialog", Font.BOLD, 12));
        txt_mess.setBackground(new Color (255,255,255));
        add(txt_mess);
        bt_quit = new java.awt.Button("Quitter");
        bt_quit.reshape(450,280,86,20);
        add(bt_quit);
        setResizable(false);
        bt_toc.disable();
        bt_stop.disable();
        bt_chercher.disable();
        //}}
    } //fin de la methode d initialisation

    public Cherche (Frame parent, String title, boolean modal)
    {
        this(parent, modal);
        setTitle(title);
    }

    //{{DECLARE_CONTROLS
    java.awt.Label lbl_url_in;
    java.awt.TextField txt_url_in;
    java.awt.Button bt_rechercher;
    java.awt.Label lbl_contenu;
    java.awt.TextArea txt_contenu;
    java.awt.Button bt_effacer;
    java.awt.Button bt_toc;
    java.awt.Button bt_chercher;
    java.awt.Button bt_stop;
    java.awt.Button bt_quit;
    java.awt.TextField txt_mess;
    //}}

    public synchronized void show()
    {
        Rectangle bounds = getParent().bounds();
        Rectangle abounds = bounds();

        move(bounds.x + (bounds.width - abounds.width)/ 2,
            bounds.y + (bounds.height - abounds.height)/2);

        super.show();
    }

    Thread Ch_Thread = null;

    public void start ()

```



```

{
    Ch_Thread = null;
    Ch_Thread = new Thread(this, "Ch_T");
    Ch_Thread.start();
}

public void stop ()
{
    Ch_Thread.stop();
    Ch_Thread = null;
}

public void run ()
{
    while (Thread.currentThread() == Ch_Thread)
    {

        String ad = txt_url_in.getText();

        affichage = "Contenu de la page : " + ad + LSEP;
        String list_tit = affichage;

        try
        {
            String inputLine = "";
            String tmpLine = "";

            int debut = 0;//
            int fin = 0;//

            int deb_tit = 0;
            int fin_tit = 0;

            boolean trouve;
            boolean trouv_tit;

            String unlien = ""; //un lien trouve.
            String untitre = ""; //un titre trouve

            String critere = "href=" + "\"" + "http://";
            String crit_titres = '<' + "h";
            String crit_fin = '<' + '/' + "h";

            bt_chercher.disable();
            URL lieu = new URL(ad); //lieu est l url trouve en
                adresse
            txt_mess.setText("Connection à l URL ...");

            InputStream in_str = lieu.openStream();
            DataInputStream dis = new DataInputStream(in_str);

            txt_contenu.setText("");
            txt_mess.setText("Recherche en cours... Veuillez
                patienter...");
            bt_stop.enable();
            bt_quit.disable();
        }
    }
}

```



```

        this.stop();
    }
} //fin du while (Thread.currentThread() == Ch_Thread)
} //fin de la methode run

public boolean handleEvent(Event event)
{
    if(event.id == Event.WINDOW_DESTROY)
    {
        hide();
        return true;
    }
    if (event.target == bt_quit && event.id == Event.ACTION_EVENT)
    {
        btquit_Clicked(event);
    }
    if (event.target == bt_rechercher && event.id ==
        Event.ACTION_EVENT)
    {
        btRechercher_Clicked(event);
    }
    if (event.target == txt_url_in && event.id ==
        Event.ACTION_EVENT)
    {
        txtUrlIn_EnterHit(event);
    }
    if (event.target == bt_effacer && event.id ==
        Event.ACTION_EVENT)
    {
        btEffacer_Clicked(event);
    }
    if (event.target == bt_toc && event.id == Event.ACTION_EVENT)
    {
        btToc_Clicked(event);
    }
    if (event.target == bt_chercher && event.id ==
        Event.ACTION_EVENT) {
        btChercher_Clicked(event);
    }
    if (event.target == bt_stop && event.id == Event.ACTION_EVENT)
    {
        btStop_Clicked(event);
    }
    return super.handleEvent(event);
} // fin de la methode handleEvent

void btquit_Clicked(Event event)
{
    hide();
} // fin de la methode btquit_Clicked(Event event)

void btToc_Clicked(Event event)
{
    //txt_contenu.setText(list_tit);
} // fin de la methode btConnect_Clicked

void btRechercher_Clicked(Event event)
{

```

```

        String adresse; // 1 adresse http

        txt_mess.setText("");
        txt_contenu.setText("");
        this.start();

    } // fin de la methode btRechercher_Clicked

    void btStop_Clicked(Event event)
    {
        if (Ch_Thread != null)
        {
            this.stop();
            txt_mess.setText("Recherche interrompue par l
            utilisateur");
            bt_stop.disable();
            bt_quit.enable();
        }
        else
        {
            //System.out.println("Thread : " +
Thread.currentThread());
        }
    } // fin de la methode btStop_Clicked

    void txtUrlIn_EnterHit(Event event)
    {
        btRechercher_Clicked(event);
    } // fin de la methode txtUrlIn_EnterHit

    void btEffacer_Clicked(Event event)
    {
        txt_url_in.setText("");
    } // fin de la methode btEffacer_Clicked

    void btConnect_Clicked(Event event)
    {
        String choix;

        choix = txt_contenu.getSelectedText();
        Connect(choix);
    } // fin de la methode btConnect_Clicked

    void btChercher_Clicked(Event event)
    {
        String choix;

        choix = txt_contenu.getSelectedText();
        txt_url_in.setText(choix);
        this.start();

    } // fin de la methode btChercher_Clicked

    void Connect (String adresse)
    { /*
        AppletContext ac;
        URL lieu;
        try
        {

```



```

        lieu = new URL (adresse);
        // va a l adresse indiquee
        ac = getAppletContext();
        ac.showDocument(lieu);
    }
    catch (MalformedURLException me)
    {
        txt_mess.setText("L adresse introduite : " + adresse + " est incorrecte");
    }
    catch (IOException ioe)
    {
        txt_mess.setText("IOException: " + ioe);
    }*/
}

}

// fin de la boite de dialogue Cherche.

// Creation de la boite de dialogue Histo
class Histo extends Dialog {

    public Histo(Frame parent, boolean modal)
    {
        super(parent, modal);

        //{INIT_CONTROLS
        setLayout(null);
        addNotify();
        resize(insets().left + insets().right + 350,insets().top + insets().bottom + 296);
        labell = new java.awt.Label("Voici les pages deja parcourues :");
        labell.reshape(insets().left + 95,insets().top + 11,250,25);
        add(labell);
        txt_pages = new java.awt.TextArea(10,100);
        txt_pages.reshape(insets().left + 30,insets().top + 40,300,200);
        txt_pages.setEditable(false);
        add(txt_pages);
        bt_quitter = new java.awt.Button("Ok");
        bt_quitter.reshape(insets().left + 140,insets().top + 260,60,25);
        add(bt_quitter);
        setResizable(false);
        //}}
    }

    public Histo(Frame parent, String title, boolean modal)
    {
        this(parent, modal);
        setTitle(title);
    }

    //{DECLARE_CONTROLS
    java.awt.Label labell;
    java.awt.TextArea txt_pages;
    java.awt.Button bt go;

```

```
// Creation de la boite de dialogue Histo
class Histo extends Dialog {
```

```

public Histo(Frame parent, boolean modal)
{
    super(parent, modal);

    //{{INIT_CONTROLS
    setLayout(null);
    addNotify();
    resize(insets().left + insets().right + 350,insets().top +
insets().bottom + 296);
    label1 = new java.awt.Label("Voici les pages deja parcourues
        :");
    label1.reshape(insets().left + 95,insets().top + 11,250,25);
    add(label1);
    txt_pages = new java.awt.TextArea(10,100);
    txt_pages.reshape(insets().left + 30,insets().top +
    40,300,200);
    txt_pages.setEditable(false);
    add(txt_pages);
    bt_quitter = new java.awt.Button("Ok");
    bt_quitter.reshape(insets().left + 140,insets().top +
    260,60,25);
    add(bt_quitter);
    setResizable(false);
    //}}
}

public Histo(Frame parent, String title, boolean modal)
{
    this(parent, modal);
    setTitle(title);
}

//{{DECLARE_CONTROLS
java.awt.Label label1;
java.awt.TextArea txt_pages;
java.awt.Button bt go;

```

```

        java.awt.Button bt_quitter;
        //}}

    public synchronized void show()
    {
        Rectangle bounds = getParent().bounds();
        Rectangle abounds = bounds();

        move(bounds.x + (bounds.width - abounds.width) / 2,
            bounds.y + (bounds.height - abounds.height) / 2);

        super.show();
    }

    public boolean handleEvent(Event event)
    {
        if(event.id == Event.WINDOW_DESTROY)
        {
            hide();
            return true;
        }
        if (event.target == bt_quitter && event.id ==
            Event.ACTION_EVENT)
        {
            btquitter_Clicked(event);
        }
        return super.handleEvent(event);
    }

    void btquitter_Clicked(Event event)
    {
        hide();
    }

} // fin de la creation de la boite de dialogue Histo

```


Annexe 3 :
Pages Principales des Sites Examinés

Site Bacob

Location: <http://www.bacob.be/bacob/fr/index.html>

BACOB

Cherchez ici l'agence BACOB la plus proche de votre domicile. Toutes les possibilités de self banking chez BACOB. Vous pouvez voir ici où se trouve l'agence BACOB la plus proche de votre domicile.

Nederlandse editie


Info produits

Jcontact

Info placements

info BACOB

BACOB C'est tout bon.



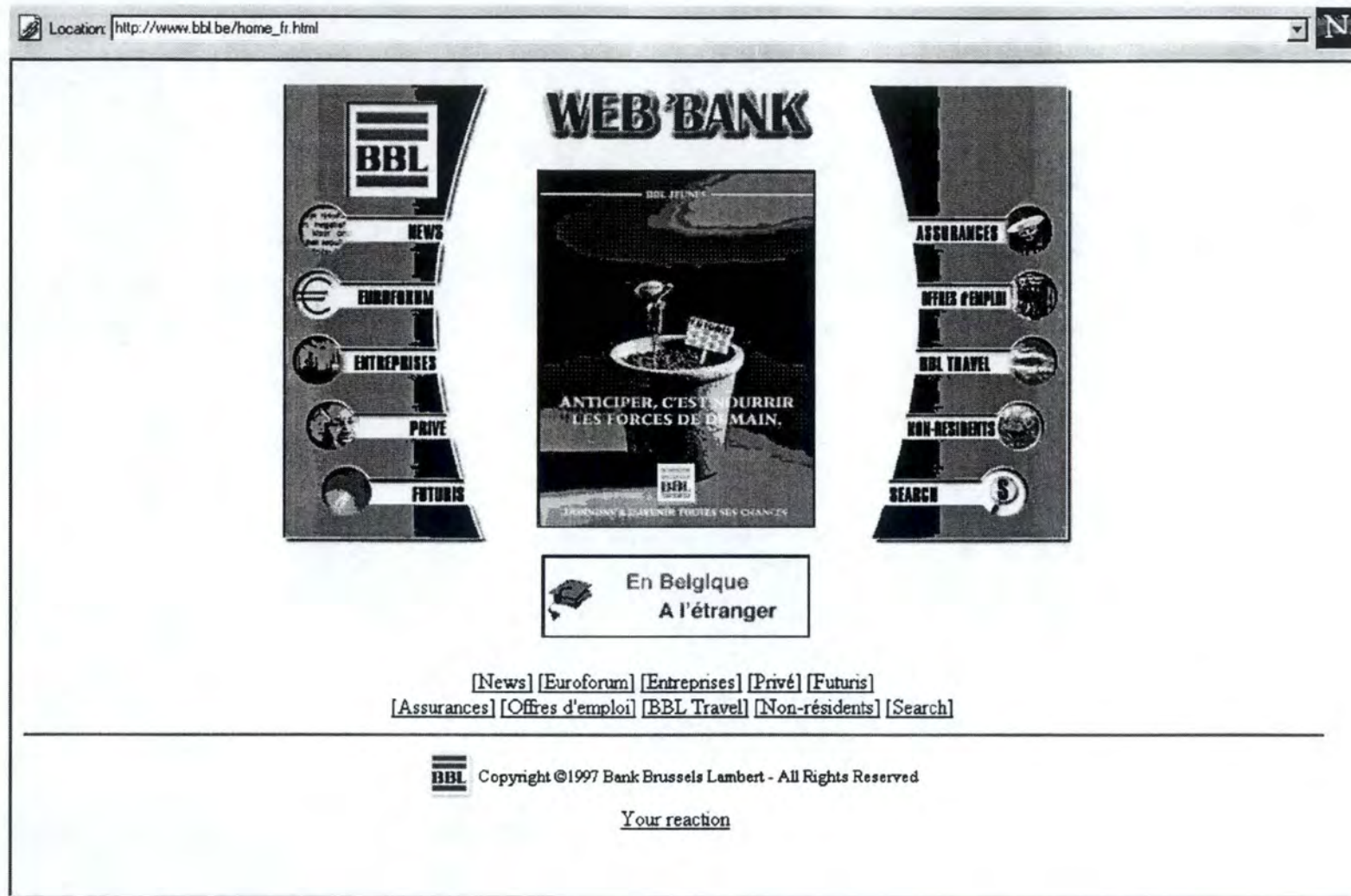
Dessin de la semaine

PROTOCOLE D'INTENTION ENTRE LE GROUPE BACOB-ARCO ET LE GROUPE PARIBAS.
Vous trouverez des précisions dans notre communiqué de presse du 5 août 1997 en consultant notre rubrique 'ACTUALITES'.

DE NOS JOURS, PEUT-ON OBTENIR UN RENDEMENT NET DE 4,25% SANS RISQUE?
CHEZ BACOB, OUI
Vous en saurez plus en consultant notre rubrique 'AFFICHE'.

OFFRE SPECIALE: DES BROCHURES D'INFORMATION GRATUITES

Site BBL



Site CGER

Location: <http://www.cger.be/>

Bienvenue

CGER On Line

RECHERCHE

Guide

Emplois

Evénements

EN EPINGLE

ES-PIX EQUITY 6

Utilisez **Netscape 3**

webmaster@cger.be

Mes études

Mes voyages

Ma voiture

Ma banque

Mon argent

Ma maison

LA BANCASSURANCE INTERACTIVE

Choisissez un module:

- Choisissez un module:
- L'Assistance Interstage
- Le Financement Auto
- Votre Assurance Incendie
- Les Prêts Hypothécaires
- Le Crédit Confort
- L'Assistance Temporaire
- Module d'orientation placements

Fortis worldwide

FILMS MUSIQUES CONCOURS

profiter de toutes les

tion et implémentation

Site Crédit Communal

The screenshot shows a web browser window with the address bar displaying "http://www.creditcommunal.be/home.html". The website header features the "Crédit Communal" logo and a slogan "ON SERA VOTRE BANQUE". A left sidebar contains a navigation menu with options like "Les informations", "Général", "Produits", "Agences", "Conjoncture", "Culture", and "Mailing list". The main content area lists several news items and services, including "Mega Assistance", "Le Guide d'Achats", "PC Banking", and "Axion Forum 2000". Logos for "axion" and "DEXIA" are visible at the bottom of the sidebar.

Location: <http://www.creditcommunal.be/home.html>

Bienvenue au Crédit Communal
Version 10

Les informations
Les informations
Général
Produits
Agences
Conjoncture
Culture
Mailing list

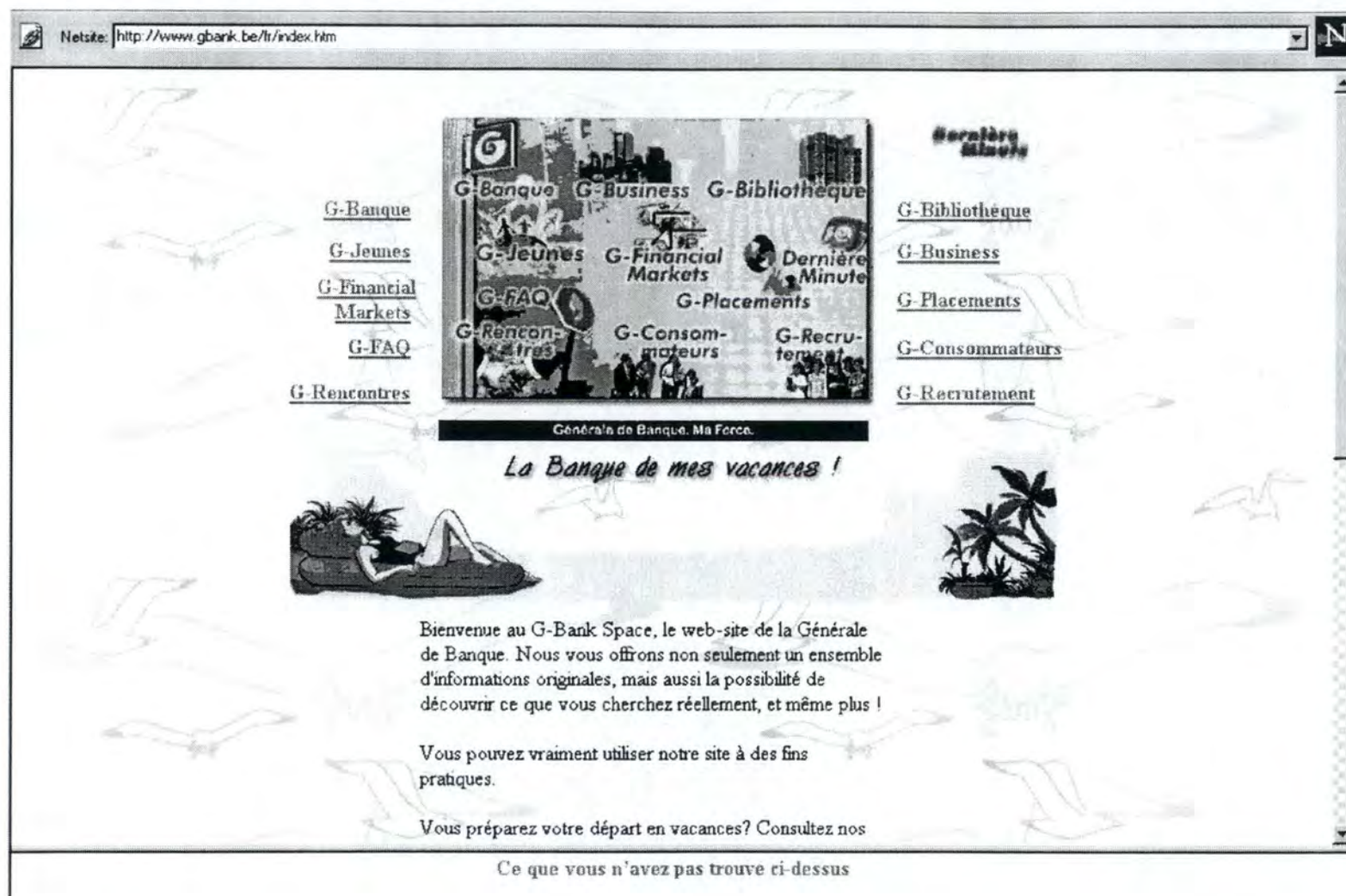
Mailez nous

axion

DEXIA

- Mega Assistance (28/07/97)
Mega Assistance s'occupe de tout. C'est une assurance-assistance qui vous procure la certitude d'une vie insouciante, à vous et à votre famille. Elle se soucie également de votre voiture, moto, caravane ou remorque. Où que vous soyez, quoi qu'il arrive. Et tout ça pour moins de 7 F par jour.
- Le Guide d'Achats (11/06/97)
Avec votre carte VISA/Eurocard du Crédit Communal ou votre carte Budgetline dans une main, et la dernière édition du Guide d'Achats dans l'autre main, vous bénéficiez de 5 à 30% de réduction dans d'innombrables magasins et restaurants. Payer moins, voilà qui est désormais simple comme bonjour.
- Devenez votre propre banquier grâce à PC Banking (28/05/97)
En complément du Phone Banking et du Self Service Banking, le Crédit Communal vous propose désormais un nouveau service pour gérer vos opérations bancaires depuis votre domicile: PC Banking. Si vous décidez sans tarder d'effectuer vos opérations bancaires à la maison depuis votre PC, vous bénéficiez de deux trimestres gratuits.
- Excellents résultats combinés du groupe Dexia en 1996 : 19,4 milliards de BEF
- Axion Forum 2000
Un forum pour tous les prophètes de malheur ou tous les créatifs entreprenants de moins de 25 ans. Le ciel nous tombera-t-il sur la tête le 31 décembre 1999 sur le coup de minuit ou sauverez-vous le monde de l'apocalypse grâce des initiatives éclatantes?
- Emplois vacants au Crédit Communal.
Nous offrons à ceux qui veulent croître avec nous un environnement de travail motivant avec de réelles possibilités d'évolution et des avantages extra-légaux.

Site Générale de Banque



Site Arcadis

Netsite: <http://www.arcadis.be/>

[Nederlandse versie](#) ◀
[English version](#) ◀

ARCADIS

25-08-97, 15:21

Bienvenue !

Arcadis
Internet Service
Provider

Arcadis est un fournisseur de services Internet situé en Belgique et prochainement au Luxembourg.

Outre les différentes possibilités de connexions à l'Internet pour les entreprises et les particuliers, Arcadis offre également divers services d'hébergement et d'aide technique à la création de sites Web.

La Grand Place de Bruxelles en direct sur Internet.

Accès Internet
Nouveau numéro d'accès

BNN

BNN est un magazine en ligne sur Bruxelles, sa cyberculture et ses loisirs. Vous y trouverez la billetterie de la FNAC ainsi que le hit-parade "le Tip Top", tous les samedis.

Ad VALVAS
www.advalvas.be

The Belgian White Pages.
Enregistrez-vous dans les pages blanches d'Internet.

st@rtet Un site destiné aux créateurs d'entreprises.

HOME Shopping Home Shopping online!
Plus de 1200 services à domicile sur Bruxelles, Liège et Anvers.

BD 33113

Un lieu de découverte et d'expressions pour tous les amateurs de Bandes Dessinées.

The 9th Dour Festival
Du 10 au 13 juillet.
Le plus important festival de la partie francophone de Belgique.

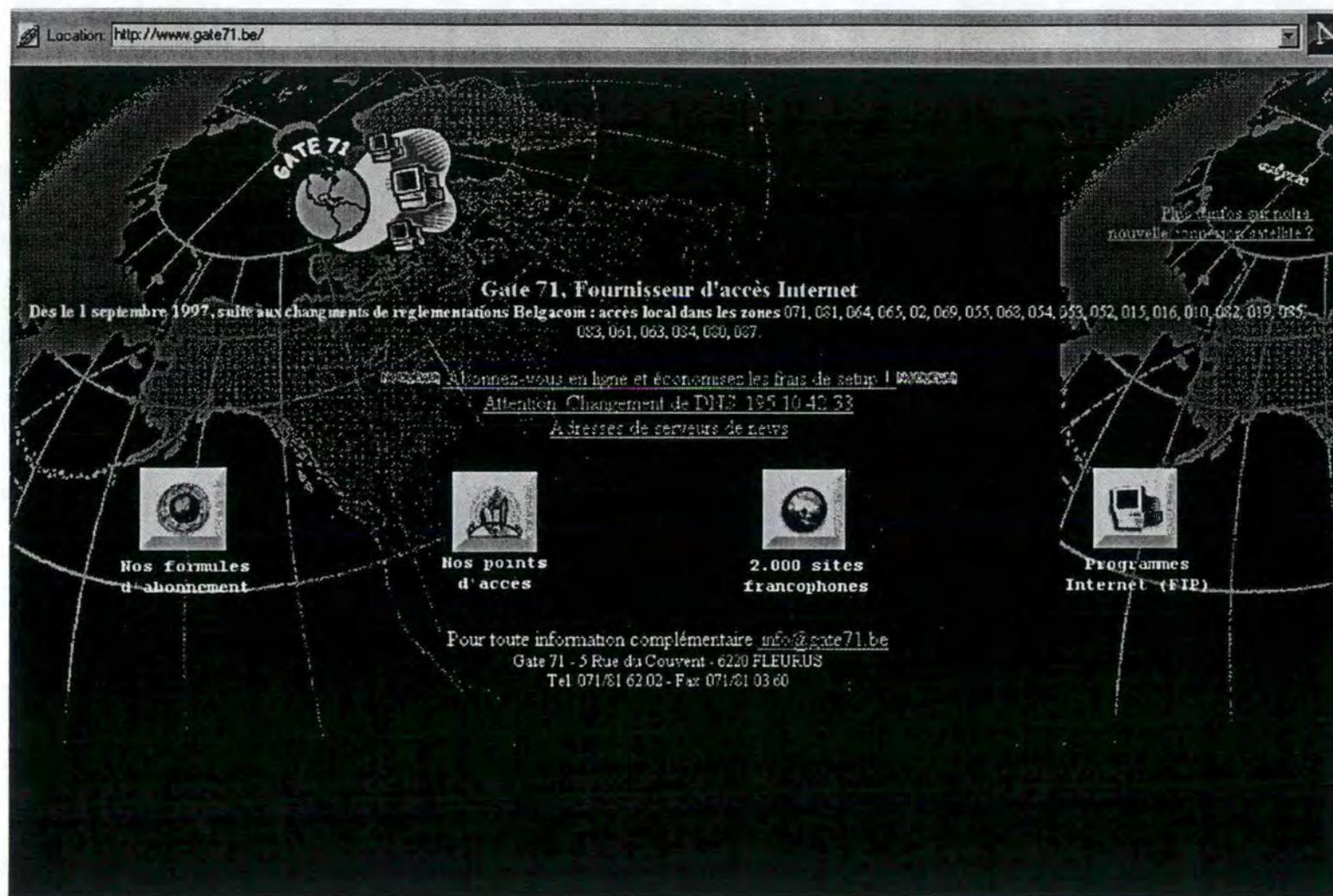
Premiers 100 Internet

L'Internet pour enfants.
Pour tous les cyber-mômes francophones et/ou francophiles.

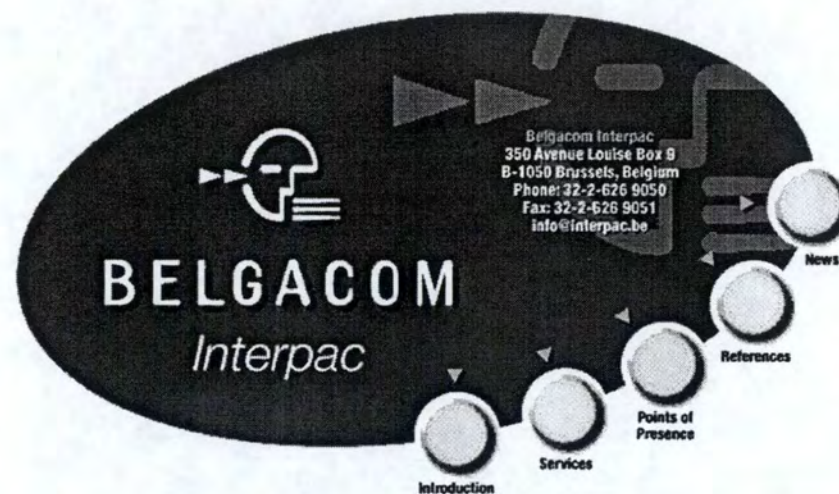
If you like and appreciate jazz, then this is a place for you.

Le Chat Bleu
est une jeune a.s.b.l. dont l'objet est essentiellement la promotion de la

Site Gate71



Site Interpac



[\[Introduction\]](#) [\[Services\]](#) [\[Points of Presence\]](#) [\[References\]](#) [\[News\]](#)

Contactadvertenties in Vlaanderen

by [IP Netverlusing](#)

Site Ping

Netsite: <http://www.ping.be/>

telemarketeers
WANTED

click here for more info

 **PING**

**THE NEW OSI
TELEPHONE
ZONES WILL
ARRIVE
SEP 1997**
**CLICK
HERE**

Your link to a world of people

The old number of POP Gent is no longer available... please only use (019 268 4

Company **Support** **Community** **Exclusive**

pluslotto.com **24 hours a day**
Supporting the International Federation of
Red Cross and Red Crescent Societies

CLICK - HERE

Tel: 070.233 772 (General Info)
Tel: 070.233 773 (Helpdesk)
Fax: 070.233 771

**Snapshot
of the Day**

1997 © PING N.V.
Webmaster@ping.be




Site Skynet



Site Abelcom

Location: <http://www.abelcom.net/abelcom/fr/index.htm>

BIENVENUE SUR



IMMOBILIER ET BATEAUX D'OCCASION EN France

Le site des agences immobilières et des particuliers

DES CENTAINES D'ANNONCES A CONSULTER, PLUS DE 10 NOUVELLES ANNONCES PAR JOUR

PLUSIEURS CENTAINES DE VISITEURS CHAQUE JOUR

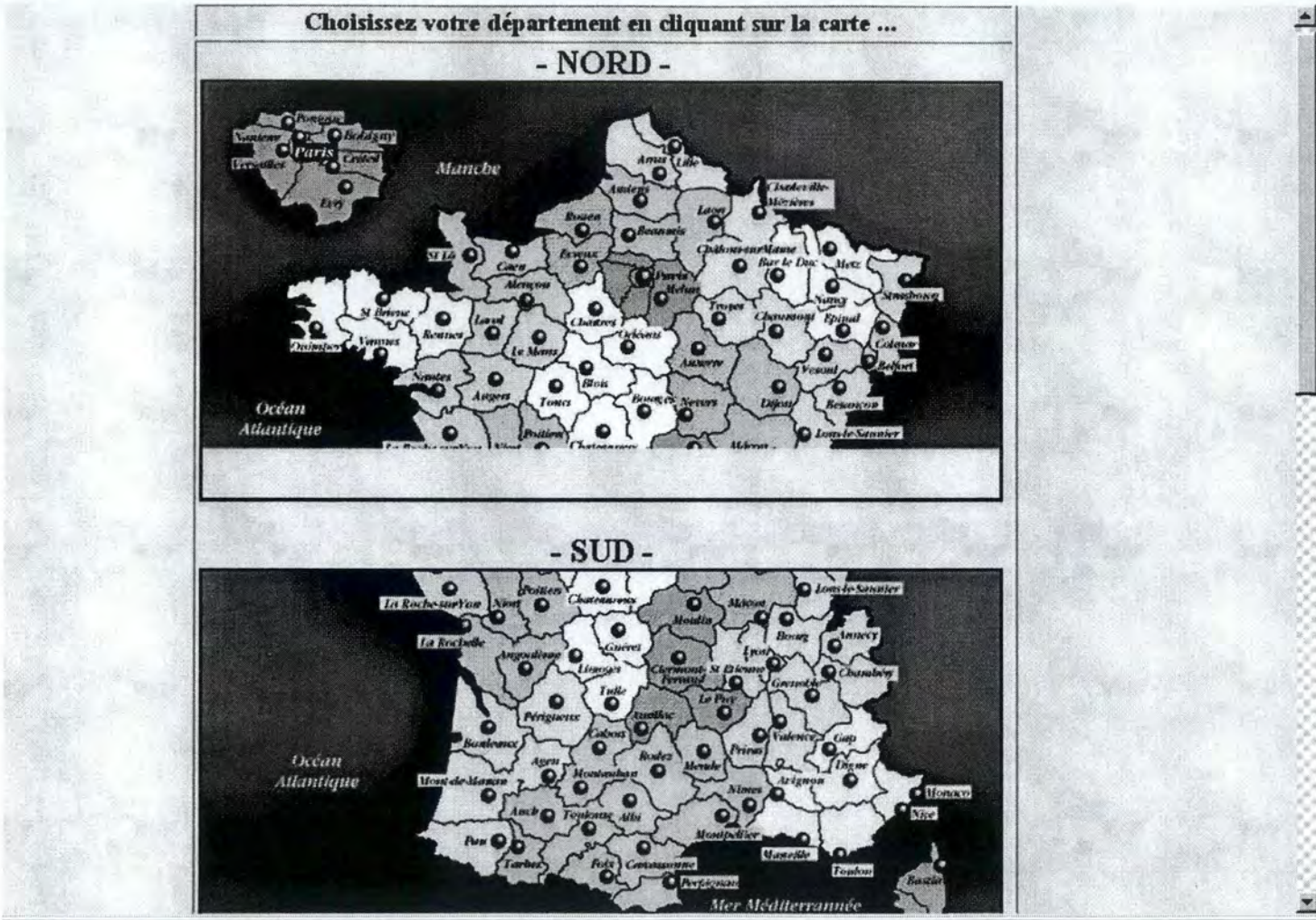
- Des annonces immobilières et bateaux d'occasions en France, classées par régions et par catégories
- L'immobilier dans le reste du Monde
- Des informations touristiques, culturelles et sportives classées par région
- Des "Adresses utiles", classées par région, où vous pourrez consulter des listes de prestataires et de services publics indispensables (Météo, hôpitaux, consulats, urgences, informations juridiques etc.)
- Pour une recherche personnalisée, si vous ne trouvez pas votre bonheur parmi toutes nos annonces
- En attendant de pouvoir acheter un appartement sur la planète MARS... Voir les photos de MARS
- Pour passer une annonce sur ABELCOM, consultez nos Tarifs

Protégez votre habitat des cambrioleurs, cliquez sur PROTEGER... Profitez de

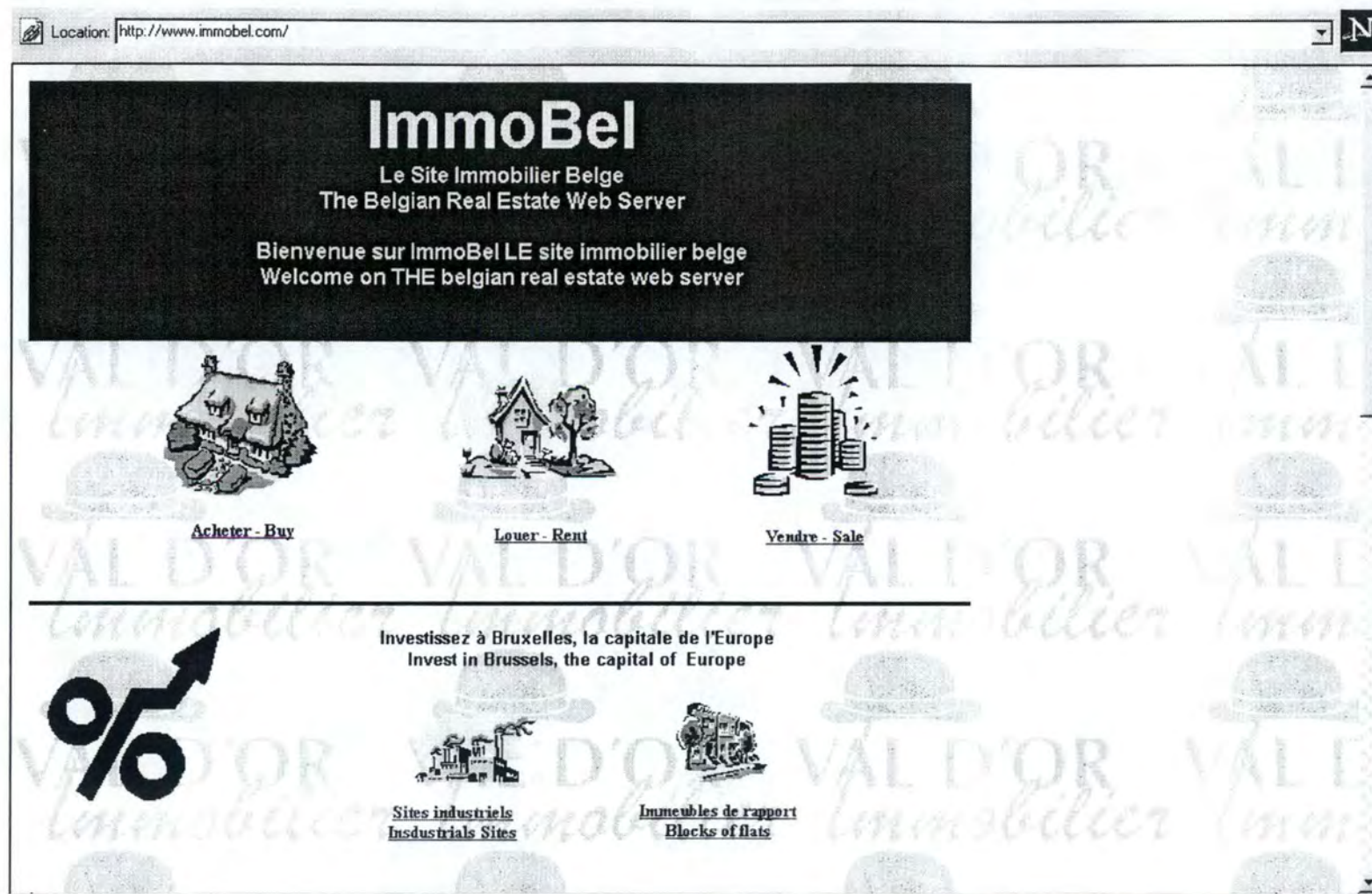
Sommaire
Annonces
Monde
Tarifs
Contacts
Recherche
Visites
Protéger
Stats
Moteurs
L'Equipe

Conception VIF

Site Cyberimmo



Site ImmoBel



Site WEB de l'Immobilier

Location: <http://immobilier.ami.fr/>



Bienvenue sur le Web de l'immobilier.

Internet Immobilier vous permet de trouver facilement et rapidement un logement à acheter ou à louer pour votre résidence principale ou pour vos vacances.
Internet Immobilier propose aussi des offres à toutes les entreprises qui cherchent des bureaux ou des locaux d'activités.

Ce Site est remis à jour quotidiennement

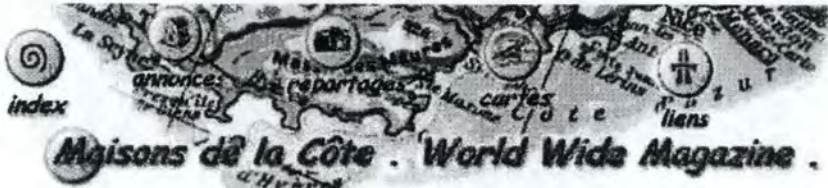
Agences **Immobilier neuf**

Toutes les annonces

	Vente	Location	
		Vide	Meublé
	Paris et sa région Province Etranger Promoteur	Paris et sa région Province Etranger	Paris et sa région Province Etranger
	Mer Montagne Campagne Etranger	Mer Montagne Campagne DOM-TOM	

Site Maisons de la Côte

Location: <http://www.easynet.fr/maisons/>



Maisons de la Côte . World Wide Magazine .

[annonces-reportages-cartes-liens](#)
Index
derniere mise à jour : 15 juillet 97

Un magazine de la Côte d'azur avec:
des [annonces immobilières](#)
des [reportages](#)
des [cartes](#) de la région
la liste des [agences immobilières](#)
4 pages de [liens](#) internationaux.
Nouveau : [commerces à vendre](#)


Service gratuit:
Vous êtes acheteur ou vendeur
d'un bien sur la Côte d'azur :
[contactez tous nos annonceurs](#)
avec un seul formulaire.



ImmoGolf Sur la Côte d'azur
terrains à bâtir,
appartements,
villas... [click](#)

Recevez gratuitement notre catalogue

Côtes de provenance



Données actualisées